PARAMETER ESTIMATION AND MODEL DISCRIMINATION IN SINGLE RESPONSE MODELS

A Thesis submitted in partial

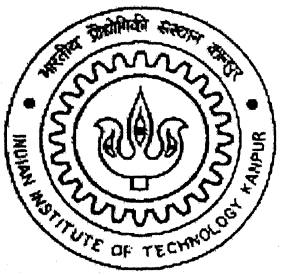
Fulfillment of the Requirements

For the Degree of

MASTER OF TECHNOLOGY

BY

RAJAT KANODIA



To

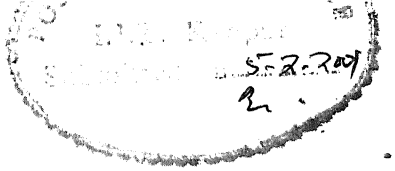
Department of Chemical Engineering

INDIAN INSTITUTE OF TECHNOLOGY, KANPUR

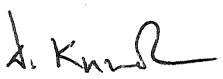
FEBRUARY, 2001

DEDICATED
TO
MY PARENTS

CERTIFICATE



This is to certify that the present work entitled **parameter estimation and model discrimination in single response models** *has been carried out by* **Rajat Kanodia** *under our supervision and it has not been submitted elsewhere for a degree.*



Dr.D.Kundu
Professor
Dept. of Mathematics
Indian Institute of Technology
Kanpur

Dr.M.S.Rao
Professor
Dept. of Chemical Engineering
Indian Institute of Technology
Kanpur

February, 2001

ACKNOWLEDGEMENT

I express my profound sense of gratitude and sincere thanks to my thesis supervisors, Dr.M.S.Rao and Dr.D.Kundu, for their constant encouragement and discerning personal and technical guidance throughout the course of this work.

I would like to express my appreciation for the help and suggestions, of my colleague, *Hasanat* has given me. My thanks to him are inexpressible.

I thank all my classmates for their stimulating conversations and good humour. It is a pleasure to reconcile the moments I have spent here in IITK with all my classmates, especially *Ankur, Dhaval, Deba, Bhola, Mandhani, Subodh, and Ghuge.*

I would also like to thank all my lab mates, *Deepak, Prashant, Siva, Sriram and Poonam*, for their moral support and help. They created cheerful atmosphere so working in the lab never seemed boring.

Finally, to my parents, my brother, and my relatives in IITK, goes my eternal gratitude for their constant love and support.

Rajat

ABSTRACT

Parameter estimation is a discipline that provides tools for the efficient use of data for aiding in mathematically modeling of phenomenon and the estimation of unknowns appearing in these models. The problem of estimating parameters is that of finding unknowns appearing in an equation describing a system.

In large majority of real systems, the investigator frequently comes across situations where several models have been postulated for the same system. This necessitates the discrimination among the competing models and selection of the best model.

One way to estimate the parameters for a large variety of models is to use least squares technique which involves minimizing the sum of squares of the differences between measurements and the model values. The function to be minimized can be either linear or nonlinear.

In the present work, data are generated for an exponential model and two methods of parameter estimation, viz; the *Simplex* method and the *Marquardt* method are applied to estimate the unknown parameters of the model. The subroutines are used for both the methods and the main programs are developed. The considered models are nonlinear in nature. The model discrimination is also carried out using two different methods, viz; the Bayesian method and the Distance method. The computations for the *Simplex* method are done in C and that for the *Marquardt* method are done in Fortran.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

NOMENCLATURE

$D_n^{(u)}$	discrimination index in the distance method
E(y)	expected value of y
I	identity matrix
k	total number of independent variables
$K_n^{(u)}$	discrepancy of model u
m	total number of models
L_i	likelihood of the $_i$-th model
$\underline{M}$	moment matrix
n	total number of experimental runs
p	total number of parameters
Q	weighting matrix
S	sum of squares of residuals
u	nx1 vector of the difference between simulated and the predicted response
V	variance-covariance matrix of response

Greek Symbols

$\underline{\varepsilon}$	vector of experimental error associated with the measurement of responses
$\underline{\eta}$	vector of predicted responses
$\eta_{n,u}$	predicted response using u-th model for n-th experiment
$\underline{\theta}$	1xp vector of parameters

$\underline{\hat{\theta}}$	an estimate of θ
$\underline{\xi}$	1xk vector of independent variables
λ	likelihood ratio
α	correction vector of parameters
σ^2	variance associated with model u

CHAPTER 1
INTRODUCTION

The main purpose of the study of a chemical or physical process is to find out how the system behaves so that we can make recommendations for its future developments. The most powerful means of doing this is to build a mathematical model for the system through which it becomes possible to predict, control and optimize the system. The models may be in the form of algebraic, differential, or integral equations.

In a kinetic information the form of the rate equation or model is not known a priori, although physiochemical insight and several formalisms limit the spectrum of possible models. Also unknown are the values of the rate constants and the adsorption equilibrium constants etc, i.e. the parameters of the model. A kinetic investigation, therefore, mainly consists of two parts: model discrimination and parameter estimation. Both tasks are evidently based upon experimental data. An example of a model is the expression for the rate of a chemical reaction between two species A and B, which may be represented by

$$-r_a = -dC_a/dt = kC_a^m C_b^n , \qquad (1.1)$$

Where: $-r_a$ is the rate of consumption of the species A, k is the rate constant, and m and n are the reaction orders.

Normally, in practical situations, one encounters more complicated models. However, in general, all phenomenon can be theoretically represented by a mathematical model of the form

$$y = \eta + \varepsilon = f(\underline{\xi},\underline{\theta}) + \varepsilon , \qquad (1.2)$$

where: η is the expected value of the dependent variable (response) y, $\underline{\theta}$ is the vector of parameters, $\underline{\xi}$ is the vector of independent variables and ε is the vector of random errors. It is often possible to postulate several meaningful models, which can describe the same system. The investigator is faced with dual problem of choosing the best among the rival models and obtaining the best estimates of the parameters involved in the selected

model. The first stage, in which the precise mathematical relationship applicable to the system is identified, is known as the specification stage and the second one, in which the precise estimates of the parameters are obtained, is known as the estimation stage. Both these constitute the important goals of modeling.

1.1 MODELING STATEGY

In some cases, when an experimenter starts with an object of modeling, he may have some knowledge about the possible mechanism while in others he may not know anything about the system. In most practical cases, normally he may have partial knowledge about the system.

If one does not have any knowledge about the system, he may resort to purely empirical modeling, while if he has complete knowledge about the system he may directly proceed with the estimation stage. In most cases, however, he may be in between these two broad aspects, which leads to the so-called mechanistic modeling. The modeling strategy may be summarized as in Fig. 1.1

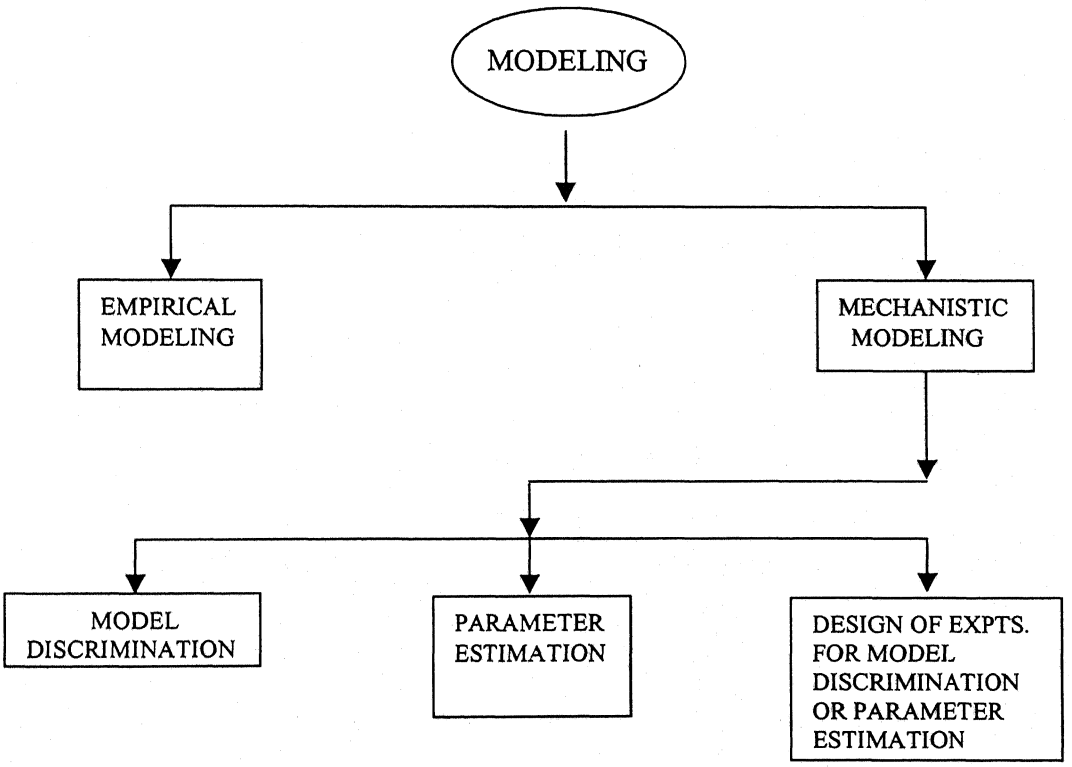


Fig.1.1 MODELING STRATEGIES OF SYSTEMS

The various modeling strategies are further discussed below.

1.1.1 EMPIRICAL MODELS

Though one would like to find the real model applicable to the physical situation, it may be invariably an elusive goal. Under these conditions, one is left with two choices, empirical modeling and mechanistic modeling. These empirical models may be polynomials in the independent variables. A typical empirical model may be represented by:

$$y_i = \theta_0 + \theta_1 \xi_{1i} + \theta_2 \xi_{2i} + ... + \theta_k \xi_{ki} + \theta_{11} \xi_{1i}^2 + ... + \theta_{kk} \xi_{ki}^2 + \varepsilon_i \,, \qquad (1.3)$$

Where, ε_i is the error associated with measurement y_i, θ and ξ refer to the corresponding parameter and independent variable respectively, i refers to the i-th measurement and k refers to the k-th independent variable in the i-th measurement. Empirical models are arbitrarily chosen on the apparent functional relationship of the response to the independent variables. Generally, either a polynomial given by equation (1.3) or a power law model given by equation (1.1) is used to represent empirical models. These have no relation, whatsoever, to the true mechanism of the system under consideration. Whenever the phenomenon under consideration is very complex, empirical models give useful guidance in predicting the response within the range of experimentation.

1.1.2 MECHANISTIC MODELS

A mechanistic model is a mathematical relationship between response (dependent variable) and the independent variables derived from a consideration of a plausible mechanism. For example, consider a simple heterogeneously catalyzed surface reaction

$$A.s + B.s \rightarrow C.s + s \,,$$

Where, A.s, B.s and C.s represent the adsorbed species A, B and C respectively and s denotes a vacant site. If the surface reaction is rate controlling, the model may be represented by

$$y = \frac{\theta_1 \xi_1 \xi_2}{\left(1 + \theta_2 \xi_1 + \theta_3 \xi_2\right)^2} \qquad (1.4)$$

Where y is the rate of reaction and ξ_1 and ξ_2 are the gas phase partial pressures of the species A and B respectively.

The use of mechanistic modeling is justified when the mechanism is fairly known. Two important aspects of mechanistic modeling consist of parameter estimation and model discrimination.

1.2 SINGLE AND MULTI RESPONSE MODELS

In general, if η is the true value of the observed response, $\underline{\xi}$ is the vector of independent variables and $\underline{\theta}$ is the vector of the unknown parameters, η is given by

$$\eta = f(\underline{\xi}, \underline{\theta}). \qquad (1.5)$$

The above equation is the true value of the observed response relation.

$f(\underline{\xi}, \underline{\theta})$ may be either a linear or a nonlinear function in the parameters, $\underline{\theta}$. If for a particular system only one response (η) can be measured, such a system is called a single response system.

However, an experimenter finds himself in a situation where, for a given set of experimental conditions, not one but several responses can be measured in a process. In general, a multi response model can be denoted by

$$y = \eta(\underline{\xi}, \underline{\theta}) + \varepsilon, \qquad (1.6)$$

Where η represents the true value of the response y, ε corresponds to the error associated with the measurement. More explicitly, the i-th response ($1 \le i \le k$) for the u-th experiment ($1 \le u \le n$) may be denoted by

$$y_u^{(i)} = \eta_u^{(i)}(\underline{\xi}_u, \underline{\theta}) + \varepsilon_u^{(i)} \qquad (1.7)$$

1.3 OBJECTIVE OF THE PRESENT WORK

One of the basic tasks of engineering and science is the extraction of information from data. Parameter estimation is a discipline that provides tools for the efficient use of data in the estimation of parameters appearing in mathematical models and for aiding in modeling of phenomena.

Earlier Lakshmi (1999) had used the package STATISTICA, which uses Gauss-Newton Method, to estimate the parameters of the models given in the section 4.3. The estimated parameters did not come near the values from which the data were generated.

It is the objective of the present study to estimate the unknown parameters of the models by reliable methods, for all the models, and then to discriminate among the rival models.

CHAPTER 2
LITERATURE REVIEW

Based on several recognized mechanisms it is often seen that various meaningful models are capable of describing a system. The task at hand before the experimenter in such a situation is to conduct a set of experiments and pick one which is most appropriate for the given system. If after conducting a set of experiments one is not able to discriminate among the various rival models and further experimentation is possible, the investigator must conduct some more experiments specific to the purpose of discrimination. Thus the task of model discrimination is accomplished sequentially. A procedure for this consists of parameter estimation, application of a discrimination criterion, and if necessary a design criterion. Various methods for estimation of parameters, model discrimination and design of experiments are available in literature e.g Mezaki et. al.(1966), Prasad and Rao(1977) etc. Some of these methods are discussed below.

2.1 MODEL SCREENING

Sometimes in the model-building process the experimenter is not confronted with just a few models but with a large number of models. For instance, in chemical engineering kinetics, the theory may suggest a large number of different courses that a reaction may take. The result may be a group of models too large to be handled effectively by model discrimination procedures. In order to reduce this number to a manageable group of models, the experimenter will need to apply preliminary screening techniques. Two widely used screening procedures are lack of fit F-test and residual analysis.

The lack of fit F-test gives an insight into the adequacy of a model. The F-statistic in model fitting is found out as the ratio of lack of fit mean square to pure error mean square. This ratio is compared to the critical value of F at the required degree of confidence and the corresponding degrees of freedom. A model is rejected when the

calculated F-ratio is greater than the critical value of F at the desired significance level. F-test is considered as a poor test for discriminating among rival models.

The residual analysis may sometimes be used for testing the model adequacy. This method primarily consists of computing the lack of fit error and making plots of residuals. When the model is adequate, the difference between the observed response and the predicted response is solely due to experimental error. Thus, plots of this error versus any independent variable should exhibit the essential characteristics of this error, such as randomness. By making such plots for all the models and comparing them, one may be able to decide which model is the most adequate.

2.2 REVIEW OF THE PREVIOUS WORK IN PARAMETER ESTIMATION

Estimation in algebraic equations, linear in parameters, is well known, and elementary computer packages contain all the associated statistical tests. Estimation in algebraic equations, which are non-linear in parameters and in differential equations, was reviewed with particular emphasis on application to kinetics, by Seinfeld(1970), Bard and Lapidus (1968), and Froment(1974). An extensive treatment of the estimation problem can be found in the books by Draper and Smith (1966). Rohatgi(1984) and Lehmann(1998) have also given useful contributions in this field.

2.2.1 OBJECTIVE FUNCTIONS

Froment(1975) has used the methods for estimating parameters in algebraic and differential equations.

Any estimation starts with the definition of a suitable objective function. Let the model equation be represented by

$$\underline{y} = f(\underline{\xi}, \underline{\theta}), \tag{2.1}$$

Where $\underline{y}$ represents the nx1 vector of responses, $\underline{\xi}$ the nxk vector of independent variables and $\underline{\theta}$ the px1 vector of unknown parameters. R experiments are carried out, and $\underline{y}$ is the measured for known $\underline{\xi}$, with certain random experimental errors. The difference between the response and the model prediction is called *residual*;

$$\varepsilon_r(\underline{\theta}) = y_r - f(\underline{\xi}_r, \underline{\theta}) \tag{2.2}$$

and the moment matrix of the residuals is given by

$$M(\underline{\theta}) = \sum_{r=1}^{R} \varepsilon_r(\underline{\theta})\varepsilon_r(\underline{\theta})^T . \tag{2.3}$$

Then the estimation aims at minimizing some functions of this moment matrix, generally called *objective function*, by the suitable choice of $\underline{\theta}$.

The most common objective function is the weighted least-square defined as

$$S = Trace[Q\, M(\underline{\theta})], \tag{2.4}$$

Where Q is an mxm weighting matrix, whose elements are selected to reflect the knowledge about the relative precision of the residuals. Q is a diagonal matrix when only sums of squares of residuals are considered and a full matrix when the sums of cross products are also taken into account.

The objective function can also be based upon the maximum likelihood principle. The likelihood function is the conditional probability relating the dependent variable $\underline{y}$ to the parameters $\underline{\theta}$.

When the variance-covariance matrix of the responses, V, is known, maximizing the likelihood is equivalent to minimizing

$$S = Trace[V^{-1} M(\underline{\theta})]. \tag{2.5}$$

When Q in equation (2.4) equals V^{-1}, the weighted least-squares estimates are also the maximum likelihood estimates. When the variance-covariance matrix of the responses is unknown, it turns out, from Bayesian analysis, that the maximum likelihood estimates are obtained from the minimization, with respect to the parameters, of

$$S = \det M(\underline{\theta}). \tag{2.6}$$

In the following sections, methods for estimation of parameters in algebraic equations are described.

2.3 ESTIMATION OF ALGEBRAIC EQUATIONS

2.3.1 LINEAR MODELS: LINEAR LEAST SQUARE TECHNIQUE

This method is applicable to linear and intrinsically linear models. It is based on minimizing the residual sum of squares. Thus, if η_k is the predicted response in k-th experiment, y_k is the corresponding observation and n experiments are performed, one may write for k-th experiment,

$$\eta_k = f(\underline{\xi}_k, \underline{\theta}) = \sum_{l=1}^{p} \theta_l \xi_{kl} \tag{2.7}$$

and for all the experiments,

$$\underline{\varepsilon} = \underline{y} - \underline{\eta} = \underline{y} - \underline{X\theta} \ , \tag{2.8}$$

Where $\underline{\varepsilon}$ is the nx1 vector of errors, $\underline{y}$ is the nx1 vector of observations, $\underline{\eta}$ is the nx1 vector of predicted responses, $\underline{X}$ is the design matrix of the order nxp. The sum of squares of errors is given by

$$\varepsilon_1^2 + \varepsilon_2^2 + \varepsilon_3^2 + \ldots + \varepsilon_n^2 = \underline{\varepsilon}^T \underline{\varepsilon} = \underline{y}^T \underline{y} - 2\underline{\theta}^T \underline{x}^T \underline{y} + \underline{\theta}^T \underline{x}^T \underline{x\theta} \, . \tag{2.9}$$

Minimizing the above equation with respect to all parameters yields the equation

$$(\underline{x}^T \underline{x})\hat{\underline{\theta}} = \underline{x}^T \underline{y} \, , \tag{2.10}$$

Where $\hat{\underline{\theta}}$ is the least squares estimate of $\underline{\theta}$. If $\underline{X}^T\underline{X}$ is non-singular

$\hat{\underline{\theta}}$ is given by

$$\hat{\underline{\theta}} = (\underline{X}^T\underline{X})^{-1}\underline{X}^T\underline{y}. \tag{2.11}$$

2.3.2 NON-LINEAR MODELS: GAUSS-NEWTON METHOD

If the functional form of the model is

$$Y = f(\underline{\xi},\underline{\theta}) + \varepsilon \tag{2.12}$$

or

$$\eta = E(y) = f(\underline{\xi},\underline{\theta}), \tag{2.13}$$

Where $f(\underline{\xi},\underline{\theta})$ is non-linear, then, for a single observation in the k-th experiment, one may write

$$y_k = f(\underline{\xi}_k,\underline{\theta}) + \varepsilon_k \tag{2.14}$$

and the error sum of squares is given by

$$S(\underline{\theta}) = \sum_{k=1}^{n} [y_k - f(\underline{\xi}_k,\underline{\theta})]^2 . \tag{2.15}$$

Since y_k and $\underline{\xi}_k$ are known, the sum of squares is a function of $\underline{\theta}$ only. The $\underline{\theta}$ are found such that the sum of squares of errors is a minimum. To find the least squares estimates $\underline{\theta}$, the function is first expanded about an initial estimate $\hat{\underline{\theta}}_0$ in Taylor's series and truncated after the first partial derivatives. The linear resultant function is,

$$f(\underline{\xi}_k,\underline{\theta}) = f\left(\underline{\xi}_k,\hat{\underline{\theta}}_0\right) + \sum_{l=1}^{p}\left[\frac{\partial f}{\partial \theta_l}\left(\underline{\xi}_k,\underline{\theta}\right)\right]_{\underline{\theta}=\hat{\underline{\theta}}_0}\left(\theta_l - \hat{\theta}_{l0}\right) \tag{2.16}$$

Abbreviating

$$f(\underline{\xi}_k,\underline{\theta}) - f(\underline{\xi}_k,\hat{\underline{\theta}}_0) = (\eta_k - \hat{\eta}_{k0}) = \hat{y}_k , \tag{2.17}$$

$$\theta_l - \hat{\theta}_{l0} = \alpha_{l0} \tag{2.18}$$

and

$$\left.\frac{\partial f(\underline{\xi}_k,\underline{\theta})}{\partial \theta_l}\right|_{\theta=\hat{\theta}_0} = z_{kl} \tag{2.19}$$

The following equation is obtained

$$\hat{y}_k = \sum_{l=1}^{p} \alpha_l z_{kl} + \varepsilon_k . \tag{2.20}$$

This is a linear approximation to Equation (2.12) and is amenable to linear least squares analysis. Considering the n observations

$$\hat{\underline{y}} = \underline{X}\underline{\alpha} + \underline{\varepsilon}, \tag{2.21}$$

Where $\underline{X}$ is the nxp matrix of partial derivatives, $\underline{\alpha}$ is the px1 correction vector, $\hat{\underline{y}}$ is the nx1 vector defined by Eq. (2.17) and $\underline{\varepsilon}$ is the nx1 vector of errors.

From the linear least squares theory, the least squares estimate $\hat{\underline{\alpha}}$, of $\underline{\alpha}$ are given by

$$\hat{\underline{\alpha}} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \hat{\underline{y}} , \tag{2.22}$$

which minimizes the sum of squares function

$$S(\hat{\underline{\theta}}) = \sum_{k=1}^{n} [y_k - f(\underline{\xi}_k, \hat{\underline{\theta}}_0) - \sum_{l=1}^{p} \alpha_l z_{kl}]^2. \tag{2.23}$$

Choosing a set of parameters $\underline{\theta}_0$, they may be recursively upgraded by computing $\hat{\underline{\alpha}}$, the correction vector. Thus the updated parameters for the (s+1)-th iteration are given by the equation

$$\hat{\underline{\alpha}}^{s+1} = \hat{\underline{\alpha}}^{s} + \hat{\underline{\alpha}}^{s} . \tag{2.24}$$

This iterative process is continues till the following criterion is satisfied

$$\left| \left(\hat{\theta}_l^{s+1} - \hat{\theta}_l^{s} \right) / \hat{\theta}_l^{s} \right| \le \delta_l , \tag{2.25}$$

Where δ_l is a small positive number.

It was found by Hartley that incrementing $\hat{\underline{\theta}}^{s}$ by full correction, $\hat{\underline{\alpha}}^{s}$ sometimes did not lead to convergence. He suggested an increment of $\Upsilon \hat{\underline{\alpha}}^{s}$ where Υ is a positive fraction for (s+1)-th iteration might be obtained by

$$\gamma = \frac{1}{2} + \frac{1}{4} S(0) - \frac{S(1)}{(S(1) - 2S(\frac{1}{2}) + S(0))} , \tag{2.26}$$

Where S(0) is the sum of squares corresponding to $\hat{\underline{\theta}}^{s}$, S(1) is the sum of squares corresponding to $(\hat{\underline{\theta}}^{s} + \hat{\underline{\alpha}}^{s})$ and S(1/2) is the sum of squares corresponding to $(\underline{\theta}^{*s} + \frac{1}{2}\hat{\underline{\alpha}}^{s})$, where $\hat{\underline{\theta}}^{s}$ are the parameters at the end of s-th iteration and $\hat{\underline{\alpha}}^{s}$ are the corresponding corrections.

2.3.3 NUMERICAL METHODS

As mentioned already, estimation in algebraic equation, which are non-linear in the parameters generally requires extensive iteration involving some hill descending procedures. The numerous procedures, which have been proposed, can be classified into three categories. The first category could be called function methods, the second is well known under the name gradient methods, and the third group the Newton methods. The basic procedures in these categories are, respectively, the univariable, the steepest descent and the Newton-Raphson methods. The principal characteristics of these three methods are shown in Table 1.

TABLE 1: Characteristics of Basic Methods of three Categories of hill descending procedures.

Category	Required	Basic procedure	Iteration cycle
Function	S	Univariable	$\theta^{(i+1)} = \theta^{(i)} + \sum_{j=1}^{p} \lambda^{0}_{je_j}$
Gradient	S and $\frac{\partial S}{\partial K}$	Steepest descent	$\theta^{(i+1)} = \theta^{(i)} - \lambda^{0}_{i}\left(\frac{\partial S}{\partial \theta}\right)_{\theta^{(i)}}$
Newton	S, $\frac{\partial S}{\partial K}$ and $\frac{\partial^2 S}{\partial k^2}$	Newton-Raphson	$\theta^{(i+1)} = \theta^{(i)} - \left[\left(\frac{\partial^2 S}{\partial \theta^2}\right)_{\theta k^{(i)}}\right]^{-1}\left(\frac{\partial S}{\partial \theta}\right)_{\theta^{(i)}}$

In the univariable procedure, the search directions are parallel to the axes, λ_j^0 is the value of the scalar λ_j corresponding to the minimum of S in the j direction in the cycle leading from $\theta^{(i)}$ to the next iteration $\theta^{(i+1)}$. The steepest descent procedure calculates S in the opposite direction of the gradient and changes direction when a minimum is attained. λ_i^0 is the value of λ corresponding to the minimum of S in the direction followed between the i+th and (i+1)-th iteration.

The Newton-Raphson procedure develops the objective function $S[\theta^{(i+1)}]$ in a Taylor series around a starting value $\theta^{(i)}$ which contains no terms beyond second order. $S[\theta^{(i+1)}]$ is then contained as the minimum of the objective function, so that a system of p linear equations is obtained, from which $\theta^{(i+1)}$ is easily calculated. The method generates simultaneously the search direction and the distance in that direction. The method converges quite rapidly in the vicinity of the minimum of S, but it is more complicated, since it requires the calculation of second order derivatives and the inversion of matrices. For quadratic objective functions, this method leads to the minimum of S in one single step.

The univariable and the steepest descent methods do not account for information gathered in previous steps and therefore may converge slowly.

2.4 REVIEW OF THE PREVIOUS WORK IN MODEL DISCRIMINATION

The problem of selection of the most appropriate model from amongst the proposed ones has been widely discussed in the literature. Almost all the various procedures proposed in the literature for discriminating among rival models use the concept of divergence in one or the other. The entire work in this field can be summarized under two broad categories; viz., the Bayesian and the Non Bayesian methods.

The nonBayesian methods include likelihood ratio methods, intrinsic parameter methods and the non-intrinsic parameter methods. The methods described in this section can be used for model discrimination in both single response models as well as multi response models.

2.4.1 BAYESIAN METHODS

The Bayes' theorem provides a useful means discriminating among rival models. Bayes' theorem is given by

$$P(A_i/B)= \frac{P(A_i)P(B/A_i)}{\sum_{i=1}^{m} P(B/A_i)P(A_i)} \qquad (2.27)$$

Where A_i (i=1,1.... m) denotes the i-th model, B denotes the data, p(Ai) denotes the prior probability of the i-th model, and P(B/Ai) denotes the likelihood of the i-th model. This is the most generally used method for model discrimination. The likelihood for any model can be obtained from knowledge of the error structure.

The work of Prasad and Rao (1977) is one of the few attempts made to improve upon the procedure for Bayesian model discrimination. According to them, more accurate and faster discrimination may be achieved if the expected likelihood is used in place of the point likelihood for calculation of posterior probabilities of models. This not only results in a sharp discrimination but also selects better points through the design criterion function. The development of such an alternative expression for the model likelihood is based on the simple idea that likelihood, being a function of the parameters, is a random variable and can, therefore, be subjected to the expectation operator.

2.4.2 NON-BAYESIAN METHODS

Non-Bayesian methods are essentially likelihood ratio methods. To illustrate the likelihood ratio approach, assume the i-th model at the u-th setting of independent ξ_u to be represented by the functional relationship.

$$\eta_{iu} = f(\xi_u, \hat{\underline{\theta}}_i) \qquad (2.28)$$

Let y denote the vector observations, which are correlated with a variance covariance matrix $\underline{V}\sigma^2$. Assume that n observations are available. The maximum likelihood function for the i-th model is given by Rao and Iyengar (1984).

$$L_i\left(\hat{\underline{\theta}},\sigma^2\right)=\frac{(n)^{n/2}}{(2\pi)^{n/2}M_i^{n/2}\sqrt{|V|}}\exp(-n/2)), \tag{2.29}$$

Where M_i is the weighted sum of squares.

It is assumed that the errors are such that

$E(\varepsilon^{(i)})=0$ for all i,

$E(\varepsilon^{(i)},\varepsilon^{(j)})=0$ for all i,j

The ratio of maximum likelihood for two models i and j, also known as the likelihood ratio, is a comparison of how well the two models fit the data.

$$\lambda=\frac{(L_i)_{max}}{(L_j)_{max}}=\left(\frac{M_j}{M_i}\right)^{n/2} \tag{2.30}$$

The likelihood ratio method of discriminating several rival models comprises of finding the likelihood ratio between the best model and the other models taken one at a time.

CHAPTER 3
METHODOLOGY

This chapter is concerned with methods used in parameter estimation and model discrimination in the present study.

3.1 METHODOLGY IN PARAMETER ESTIMATION

3.1.1 EXHAUSTIVE SEARCH

One of the procedures for extremizing a function is exhaustive search Beck and Arnold (1977). To illustrate this procedure we have considered one unknown parameter β. The function to be minimized is given by

$$S = \sum_{i=1}^{i=n} (y_i - \eta_i)^2 . \qquad (3.1)$$

$$\eta_i = f(\underline{x}_i, \underline{\beta}).$$

The best value of the parameter is the one, which gives minimum value of S. Instead of selecting only one initial estimate, a region of β is chosen in which the minimum value of S lies. Suppose that the parameter β is known to be between 0.5 and 2.25. In an exhaustive search S is calculated at equally spaced values of b in this region (β is the true value and b is an estimated value). Fig.3.1a shows S for Δb intervals of 0.25. The best $b^{(i)}$ value as indicated by Fig. 3.1a is b = 1. A more accurate value of b is then found out by conducting an exhaustive search with a smaller Δb in the region between b = 0.75 and b = 1.25.

The exhaustive search procedure is undoubtedly expensive but it does have the potential of revealing local minima in addition to the global minimum.

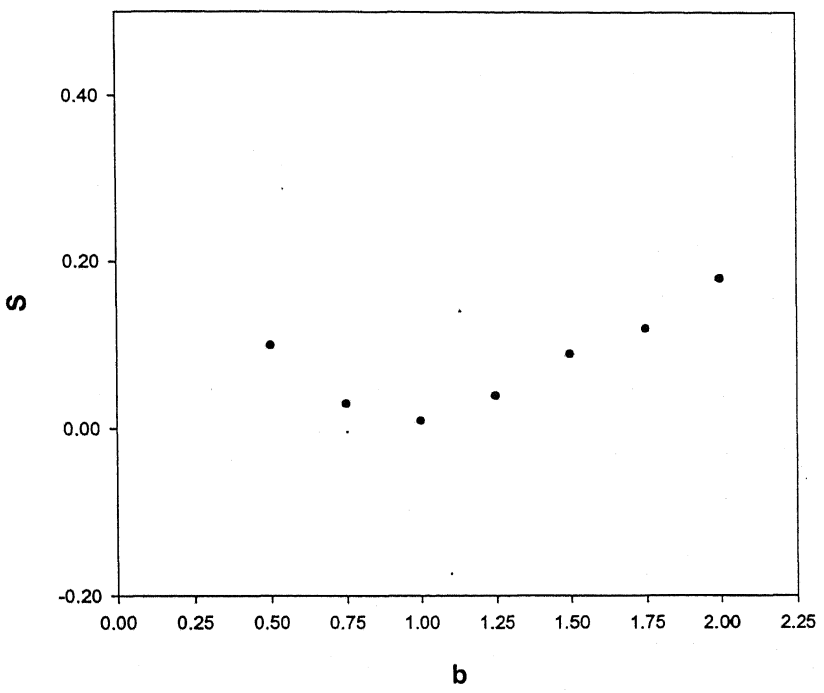


Fig 3.1 Sum of squares in an exhaustive search procedure

Suppose there are two parameters with respect to which we have to minimize the function then the search can be carried out keeping one of the parameters fixed and varying the other parameter in some data range and vice versa. This should be done for a few parameters i.e. for a few values of one parameter when we are varying the other parameter and then for a few values of the second parameter when we are varying the first parameter.

3.1.2 EXTREMIZATION OF FUNCTION

3.1.2.1 SIMPLEX METHOD

This method is used for multidimensional minimization, that is, finding the minimum of a function of more than one independent variable. The *simplex method* is due to Nedler and Mead (1965). The method requires only function evaluations, not derivatives.

A *simplex* is the geometrical figure consisting, in N dimensions, of N+1 points (or vertices) and all their interconnecting line segments, polygon faces, etc. In two dimensions the simplex is a triangle, in three dimensions it is a tetrahedron and so on.

The minimization of a function of n variables, without constraints is being considered. $P_0, P_1, \ldots, P_n$ are the (n+1) points in an n-dimensional space defining the current simplex. If one of these points is taken as the initial starting point P_0, then the other N points can be taken as

$$P_i = P_0 + \lambda\, \mathbf{e}_i, \tag{3.2}$$

Where the e_i's are N unit vectors, and where λ is a constant, which is the guess of the problem's characteristic length scale. In the present study its value is taken to be 1.0

We write y_i for the function value at P_i, and define

h as the suffix such that $y_h = \max_i(y_i)$ [h for "high"]

and

l as the suffix such that $y_l = \min_i(y_i)$ [l for "low"]

Now $\overline{P}$ is defined as the centroid of the points and $|P_iP_j|$ is the distance from P_i *to* P_j.

At each stage in the process P_h is replaced by a new point; the three operations used are- reflection, contraction, and expansion. These are defined below.

The reflection of P_h is denoted by P^*, and its co-ordinates are defined by the relation

$$P^* = (1+\alpha)\overline{P} - \alpha P_h, \tag{3.3}$$

Where α is a positive constant, the reflection coefficient. Thus P^* is on the line joining $P_h and \overline{P}$, on the far side of $\overline{P}$ from P_h with

$$\left|P^*\overline{P}\right| = \alpha\left|P_h\overline{P}\right|. \tag{3.4}$$

If y^* lies between y_h and y_l, then P_h is replaced by P^* and the method is restarted with new simplex. If $y^* < y_l$, i.e. if the reflection has produced a new minimum, then P^* is expanded to P^{**} by the relation

$$P^{**} = \gamma P^* + (1-\gamma)\overline{P}. \tag{3.5}$$

The expansion coefficient γ, which is greater than unity, is the ratio of the distance $\left|P^{**}\overline{P}\right|$ to $\left|P^*\overline{P}\right|$. If $y^{**} < y_l, P_h$ is replaced by P^{**} and the process is restarted, but if $y^{**} > y_l$ then there is a failed expansion and P_h is replaced by P^* before restarting the process.

If on reflecting P to P^* $y^* > y_l$ for all $i \neq h$ i.e. replacing P by P^* leaves y^* the maximum, then a new P_h is defined which is equal to the old P_h or P^*, whichever has the lower y value, and form

$$P^{**} = \beta P_h + (1-\beta)\overline{P}. \tag{3.6}$$

The contraction coefficient β lies between 0 and 1 and is the ratio of the distance $\left|P^{**}\overline{P}\right|$ to $\left|P\overline{P}\right|$. We then accept P^{**} for P_h and restart, unless $y^{**} > \min(y_h, y^*)$, i.e. the contracted point is worse than the better of P_h and P^*. For such a failed contraction all the P_i's are replaced by $((P_i + P_l)/2$ and the process is restarted.

The coefficients α, β, γ give the factor by which the volume of the simplex is changed by the operations of reflection, contraction or expansion respectively.

A final point concerns the criterion used for halting the procedure. The criterion adopted is concerned with the variation in the y values over the simplex. The form chosen is to compare the " standard error" of the y's in the form

$$\sqrt{\left\{\sum\left(y_i-\bar{y}\right)^2/n\right\}} \qquad (3.7)$$

with the preset value, and to stop when it falls below this value. The complete method is given as a flow diagram in Fig 3.2

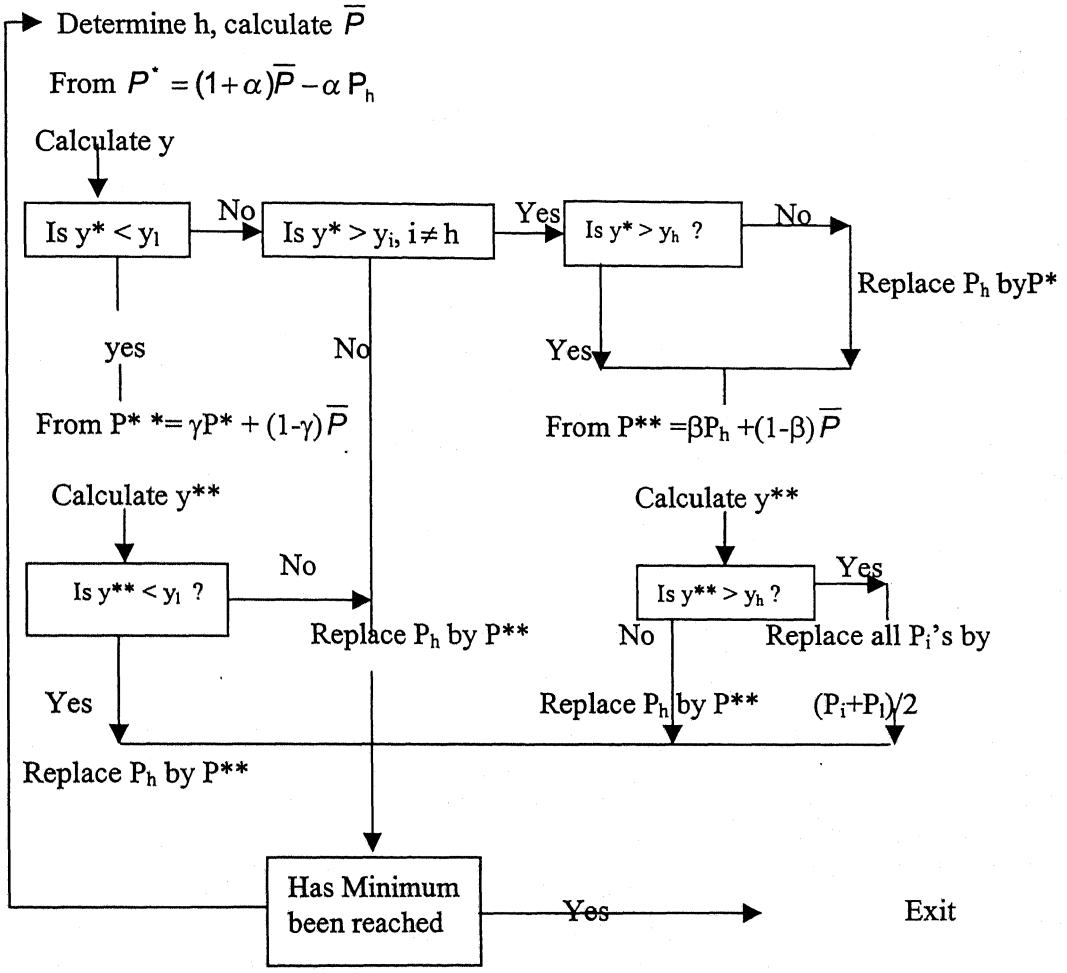


Fig 3.2 Simplex Method Logic Diagram

3.1.2.2 MARQUARDT METHOD

Marquardt (1963) gave the method described below.

Statement of The Problem: The model to be fitted to the data is

$$E(y) = f(x_1, x_2, \ldots, x_k; \theta_1, \theta_2, \ldots, \theta_m) \quad (3.8)$$
$$= f(\underline{x}, \underline{\theta}),$$

Where $x_1, x_2, \ldots, x_k$ are independent variables, $\theta_1, \theta_2, \ldots, \theta_m$ are the population values of m parameters, and E (y) is the expected value of the dependent variable y. The data points are denoted by

$$(y_i, x_{1i}, x_{2i}, \ldots\ldots, x_{ki}), \quad i = 1,2,\ldots\ldots,n. \quad (3.9)$$

The problem is to compute those estimates of the parameters that will minimize

$$\phi = \sum_{i=1}^{n} \left[y_i - \hat{y}_i \right]^2 \quad (3.10)$$
$$= \left\| \underline{y} - \underline{\hat{y}} \right\|^2,$$

Where $\hat{y}_i$ is the value of y predicted by (3.2) at the ith data point.

The model is linearized by expanding y_i^* in Taylor series about the current trial values for the coefficients and retaining the linear terms only,

$$y^*_i = \hat{y}^*_i + \left[\frac{\partial \hat{y}_i}{\partial \hat{\theta}_1} \right]^* \Delta\hat{\theta}_1 + \left[\frac{\partial \hat{y}_i}{\partial \hat{\theta}_2} \right]^* \Delta\hat{\theta}_2 + \ldots\ldots + \left[\frac{\partial \hat{y}_i}{\partial \hat{\theta}_M} \right]^* \Delta\hat{\theta}_M, \quad (3.11)$$

Where $\Delta\hat{\theta}_j = \left[\hat{\theta}_j - \hat{\theta}_j^* \right]$, $\quad j=1,2,\ldots\ldots,M.$

The asterisk designates that the quantities are evaluated at the initial trial values.

The linearized model is substituted into the objective function and the normal equations formed by setting the partial derivatives of the objective function with respect to each coefficient equal to zero:

$$\frac{\partial \phi}{\partial \hat{\theta}_j} = 0, \qquad j = 1,2,\ldots..,M.$$

Thus $\Delta\theta$ can be found by solving

$$A\,\Delta\theta = g$$

Where $A^{[m\times x]} = P^t P$,

$$P^{[n\times m]} = \left(\frac{\partial \hat{y}_i}{\partial b_j} \right), \quad i = 1,2\ldots,n;\ j = 1,2,\ldots,m,$$

$$g = \left(\sum_{i=1}^{n} \left(y_i - \hat{y}_i \right) \frac{\partial \hat{y}_i}{\partial \theta_j} \right).$$

The resulting normal equations will be of the form

$$(\underline{\theta}^t \underline{\theta} + \lambda x \underline{I})\underline{\Delta\hat{\theta}} = \underline{\theta}^t (\underline{y} - \underline{\hat{y}}^*), \tag{3.12}$$

Where I is an identity matrix of order m × m and λ is a factor that is added to the main diagonal of the $\underline{\theta}^t \underline{\theta}$ matrix. The rules for calculating λ are discussed in the original article by Marquardt (1963).

$$\underline{\theta} = \begin{bmatrix} \frac{\partial \hat{y}_1}{\partial \hat{\theta}_1} & \frac{\partial \hat{y}_1}{\partial \hat{\theta}_2} & \ldots\ldots\ldots & \frac{\partial \hat{y}_1}{\partial \hat{\theta}_M} \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \cdot & \cdot & & \cdot \\ \frac{\partial \hat{y}_N}{\partial \hat{\theta}_1} & \frac{\partial \hat{y}_N}{\partial \hat{\theta}_2} & & \frac{\partial \hat{y}_N}{\partial \hat{\theta}_M} \end{bmatrix}^*$$

$$\underline{\Delta\theta} = \begin{bmatrix} (\hat{\theta}_1 - \hat{\theta}_1^*) \\ (\hat{\theta}_2 - \hat{\theta}_2^*) \\ \cdot \\ \cdot \\ \cdot \\ (\hat{\theta}_M - \hat{\theta}_M^*) \end{bmatrix}, \qquad (\underline{y} - \hat{\underline{y}}^*) = \begin{bmatrix} \left(y_1 - \hat{y}_1^*\right) \\ \left(y_2 - \hat{y}_2^*\right) \\ \cdot \\ \cdot \\ \cdot \\ \left(y_N - \hat{y}_N^*\right) \end{bmatrix}$$

The $\hat{\underline{\Delta\theta}}$ vector and ϕ will approach zero as convergence is achieved. If convergence is achieved the final coefficients are calculated from

$\hat{\theta}_j = \hat{\theta}^* + \Delta\hat{\theta}_j$, j = 1,2,…., M.

The convergence criteria used for halting the procedure is given by

$$\frac{\left|\Delta\hat{\theta}_j\right|}{\left|\hat{\theta}_j\right|} < \gamma , \tag{3.13}$$

Where γ is some suitably small value greater than zero.

If convergence is not achieved $\hat{\underline{\theta}}^*$ is updated by replacing the old values by the new values and the process repeated.

A flow sheet illustrating the above procedure is given in Fig. 3.3

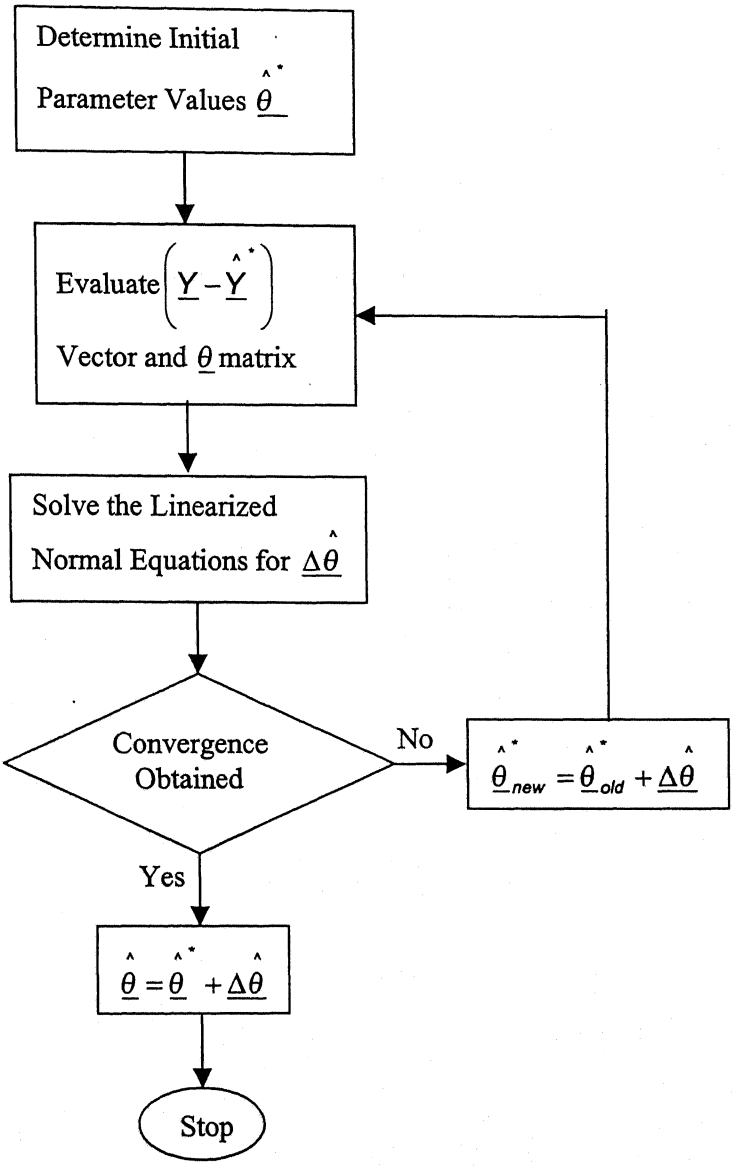


Fig 3.3 *Marquardt* Method Logic diagram

3.2 METHODOLOGY IN MODEL DISCRIMINATION

3.2.1 BAYESIAN METHOD

The Bayes' theorem provides a useful means of discriminating among rival models. Bayes' theorem states that

$$P(A_i/B)=\frac{P(A_i)P(B/A_i)}{\sum_{i=1}^{r}P(B/A_i)P(A_i)}, \tag{3.14}$$

Where A_i (i = 1,2,.....,r) denotes the i-th model, B denotes the data, $P(A_i)$ denotes the prior probability of the i-th model and $P(B/A_i)$ denotes the likelihood for the i-th model.

The Bayes' theorem requires knowledge of the prior probabilities of the various models. If one does not have knowledge of the prior probabilities of the various models, he can assign equal probabilities to all the models.

The likelihood of the i-th model can be calculated by the following expression

$$L(\underline{\theta}^{(i)}|\sigma,y)=\frac{1}{(\sqrt{2\pi}\sigma)^n}\exp\left\{-\frac{1}{2\sigma^2}(y-\eta)^T(y-\eta)\right\}, \tag{3.15}$$

Where y is the observed value of the response and η is the expected or the true value of the response. It is assumed that the errors are normally distributed and also

$E(\varepsilon^{(i)})=0$ for all i,

$E(\varepsilon^{(i)},\varepsilon^{(j)})=0$ for all i, j.

3.2.2 DISTANCE METHOD

So far as the decision on the relative adequacy of models at the n-th stage is concerned, a discrimination index (Singh, 1986) is proposed as

$$D_n^{(u)}=\frac{D_{n-1}^{(u)}.K_n^{(u)}}{\sum_{v=1}^{m}D_{n-1}^{(v)}.K_n^{(v)}}, \tag{3.16}$$

Where u denotes the model (u=1,2,..,m) $D_{n-1}^{(u)}$ is the value of the statistic for model u at the previous (n-1)-th stage and $K_n^{(u)}$ measures the discrepancy of the model in explaining the mechanism of the given process at the current, n-th stage. Since in model discrimination problems it is assumed that all the proposed models are equally plausible, it may be assumed that $D_{n-1}^{(u)}$=1/m, u=1,2,.,m.

In this method, the model, which has smaller value of the discrimination index is considered as the best one. This assessment is based on the distance between the corresponding populations. The decision on the termination of the sequential procedure can normally be taken on the basis of subjective judgement, i.e., through a comparison of the values attained by the discrimination index at a particular stage. It is proposed to stop the discrimination at the stage n*, if

$$\text{ABS}\left[\frac{D_{n^*}^{(u+)}}{D_{n^*}^{(u+)}+D_{n^*}^{(u^*)}}-\frac{D_{(n^*-1)}^{(u+)}}{D_{(n^*-1)}^{(u+)}+D_{(n^*-1)}^{(u^*)}}\right]\leq 0.001, \tag{3.17}$$

Where $D_{(n^*-1)}^{(u^*)}$, $D_{n^*}^{(u^*)}$ are the values of the discrimination index for the model $M^{(u^*)}$ at two consecutive stages. u^* and u^+ stand for the best and the second best models, respectively. Suppose that there are n observations $\underline{y}=(y_1,y_2,.....,y_n)$ taken from the population $\pi^{(0)}$. Correspondingly, another set $\underline{y}^{(u)}=(y_1^{(u)},y_2^{(u)},.....,y_n^{(u)})$ may be simulated through model $M^{(u)}$ and identified as a sample from the population $\pi^{(u)}$. Now the distribution of $\pi^{(0)}$ and $\pi^{(u)}$ being $N(\alpha^{(0)},\lambda^{(0)})$ and $N(\alpha^{(u)},\lambda^{(u)})$, respectively. The dissimilarity between the parent populations of the two samples is shown to be given by (Singh, 1986)

$$K(\pi^{(0)},\pi^{(u)})=\left[1-\left(\frac{4\lambda^{(0)}\lambda^{(u)}}{(\lambda^{(0)}+\lambda^{(u)})^2}\right)^{1/4}\exp\left(\frac{-(\alpha^{(0)}-\alpha^{(u)})^2}{4(\lambda^{(0)}+\lambda^{(u)})}\right)\right]. \tag{3.18}$$

A sample estimate of $K(\pi^{(0)},\pi^{(u)})$ and hence of the discrepancy of model u in explaining the mechanism of the process may thus be proposed as

$$K_n^{(u)}=\left[1-\left(\frac{4S_n^2S_{u,n}^2}{(S_n^2+S_{u,n}^2)^2}\right)^{1/4}\exp\left(-1/4\frac{(\bar{y}_n-\bar{y}_{u,n})^2}{(S_n^2+S_{u,n}^2)}\right)\right]^{1/2}, \tag{3.19}$$

Where $S_n^2=\underline{y}^T(I-\frac{1}{n}j_{nn})\underline{y}$,

$$\bar{y}_n=\frac{1}{n}\underline{y}^T j_{n1},$$

$$S_{u,n}^2=\underline{y}^{(u)^T}\left(I-\frac{1}{n}j_{nn}\right)\underline{y}^{(u)},$$

$$\bar{y}_{u,n}=\frac{1}{n}\underline{y}^{(u)T}j_{n1},$$

with j_{ab} as axb matrix of unit elements.

CHAPTER 4
RESULTS AND DISCUSSION

The methodology discussed in the previous chapter has been applied to the example given in section 4.3 and the results are compared.

4.1 DATA SIMULATION

To estimate the model parameters and then to discriminate among the rival models we need some data. There are two ways in which one can get the data - either using real life systems or by generating them by simulation. In the present study the data are generated using simulation.

4.2 PARAMETER ESTIMATION

Whether the data are obtained from an experiment or through simulation, the parameters of all the models must be estimated first before we proceed to discrimination. The parameters are estimated using the two methods described in the previous chapter. The subroutines of both the methods,viz; the *Simplex* method, Numerical Recipes in C, (1999) and the *Marquardt* method, Kuester and Mize, (1963) are used and the main programs have been developed. The complete program description of both the methods is given in Appendix I and Appendix II.

4.3 APPLICATION TO THE EXAMPLE

Consider a chemical reaction of the type

$$A \longrightarrow B$$

Depending on whether the reaction is of first, second, third or fourth order the following four rival models can be considered. Earlier Hill (1966) and Singh (1986), used these models for testing their criteria.

Model 1. $\eta^{(1)} = \exp\left\{-\theta_1^{(1)}\xi_1 \exp\left(-\theta_2^{(1)}/\xi_2\right)\right\}$

Model 2. $\eta^{(2)} = \left\{1+\theta_1^{(2)}\xi_1 \exp\left(-\theta_2^{(2)}/\xi_2\right)\right\}^{-1}$

Model 3. $\eta^{(3)} = \left\{1 + 2\theta_1^{(3)}\xi_1 \exp\left(-\theta_2^{(3)}/\xi_2\right)\right\}^{-1/2}$

Model 4. $\eta^{(2)} = \left\{1 + 3\theta_1^{(4)}\xi_1 \exp\left(-\theta_2^{(4)}/\xi_2\right)\right\}^{-1/3}$

Where $\eta^{(v)}$ is the expected concentration of A under the model v, ξ_1 is the reaction time, ξ_2 is the temperature, and $\theta_1^{(v)}$ and $\theta_2^{(v)}$ are the parameters of model v. The region of experimentation has been chosen to be: $0 \le \xi_1 \le 150$ and $450 \le \xi_2 \le 600$.

In this example model 2 (second order reaction) has been assumed to be the true model. Data, for model 2, are generated by the method described below

$y_{new} = y_{old} + R \times \sigma$,

Where R are the random numbers and σ is the standard deviation. The data are generated taking θ_1=400, θ_2=5000 and σ =0.05.

Data generated are being given in Table 4.1

TABLE 4.1: Response data using Model 2 as the correct model

Run	Input Variables		Response
N	ξ_1	ξ_2	y_n
1	25	575	.3961
2	25	475	.7232
3	125	475	.4215
4	125	575	.1297
5	150	550	.1292
6	25	525	.58
7	150	525	.186
8	150	550	.1294
9	25	525	.579

4.3.1 EXHAUSTIVE SEARCH

As described in the previous chapter an exhaustive search has been carried out to find out the region in which the minimum value of the function occurs, the function to be minimized is

$$S(\underline{\theta}) = \sum_{i=1}^{n} \left(\underline{y}_i - f(\underline{\theta},\underline{\xi}_i)\right)^2 \text{ for i =1,2,...,n.}$$

Where y_i is the observed value of the response and $f(\underline{\theta},\underline{\xi}_i)$ is the true or predicted value of the response. Since S is the function of parameters the range of the values of the parameters, in which the minimum value of the function lies, is found.

The search is carried out keeping one of the parameters fixed and varying the other and vice-versa. The search results are shown in the Figures 4.1 and 4.2.

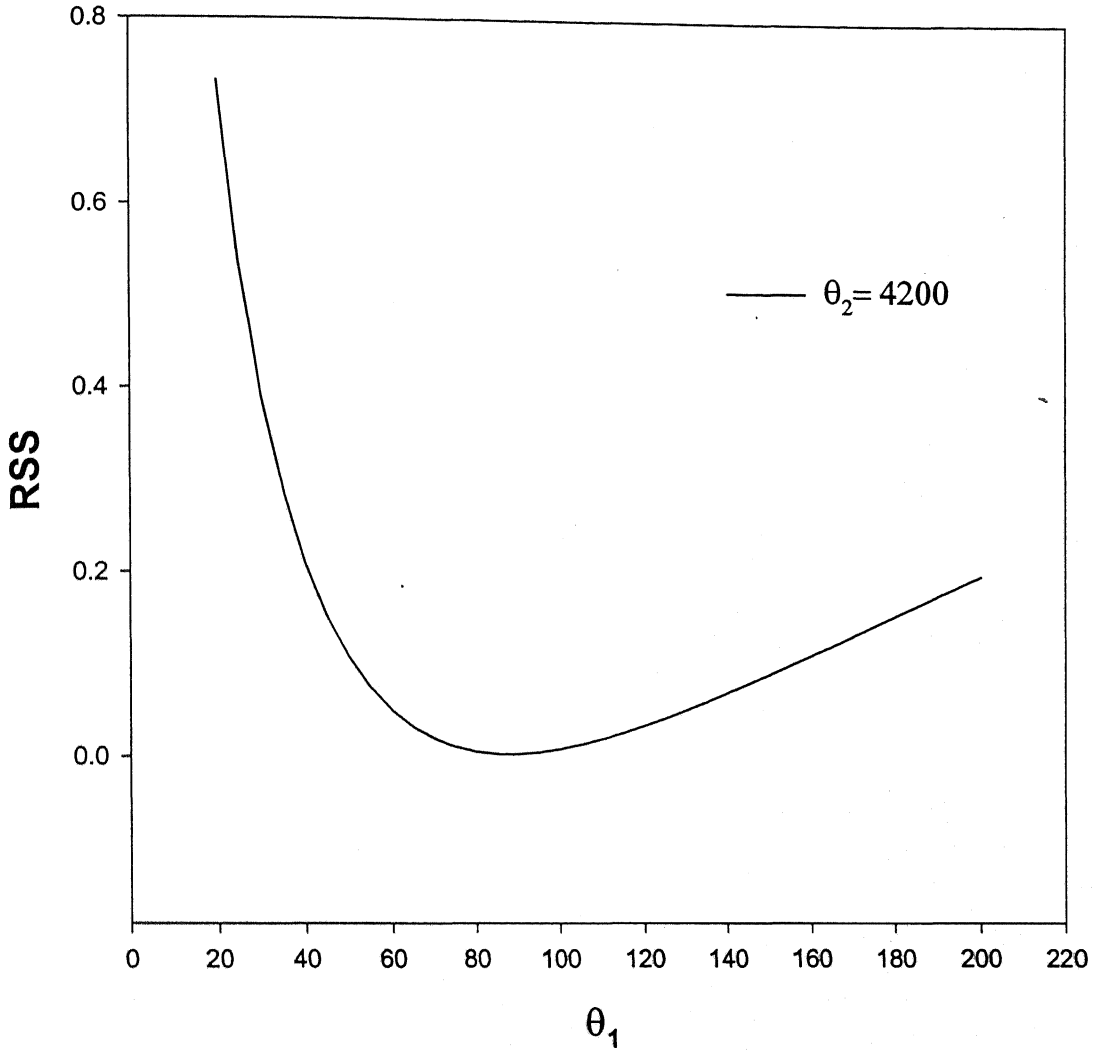


Fig. 4.1 Plot of RSS(residual sum of squares) vs θ_1

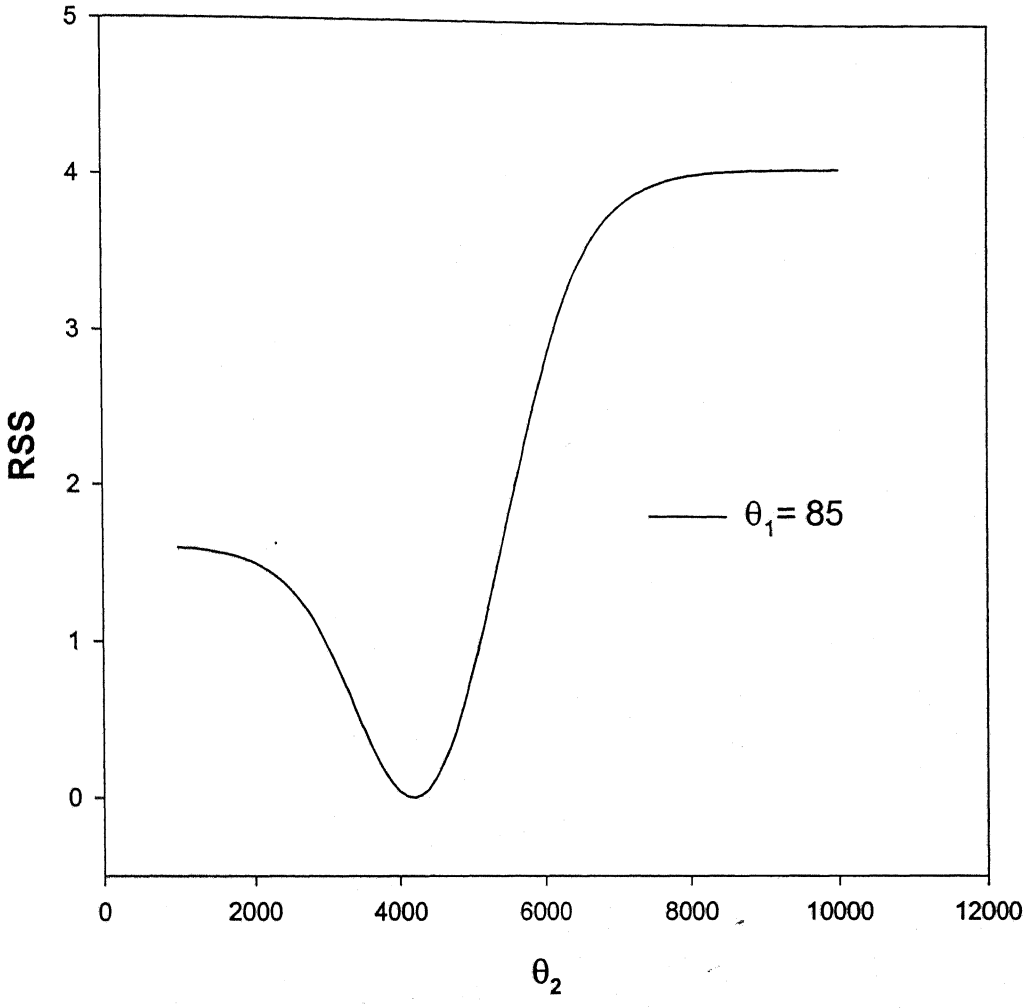


Fig. 4.2 Plot of RSS(residual sum of squares) vs θ_2

From the search results the range of the values of both the parameters found, in which the minimum value of the function lies, are as following

$$60 \leq \theta_1 \leq 120$$

and

$$3700 \leq \theta_2 \leq 5000$$

Now the area of search is concentrated within the above mentioned limits and the results are shown in Figures 4.3 and 4.4. Although the range of search is reduced for θ_2 it is not reduced for θ_1 but is done for different θ_2 values.

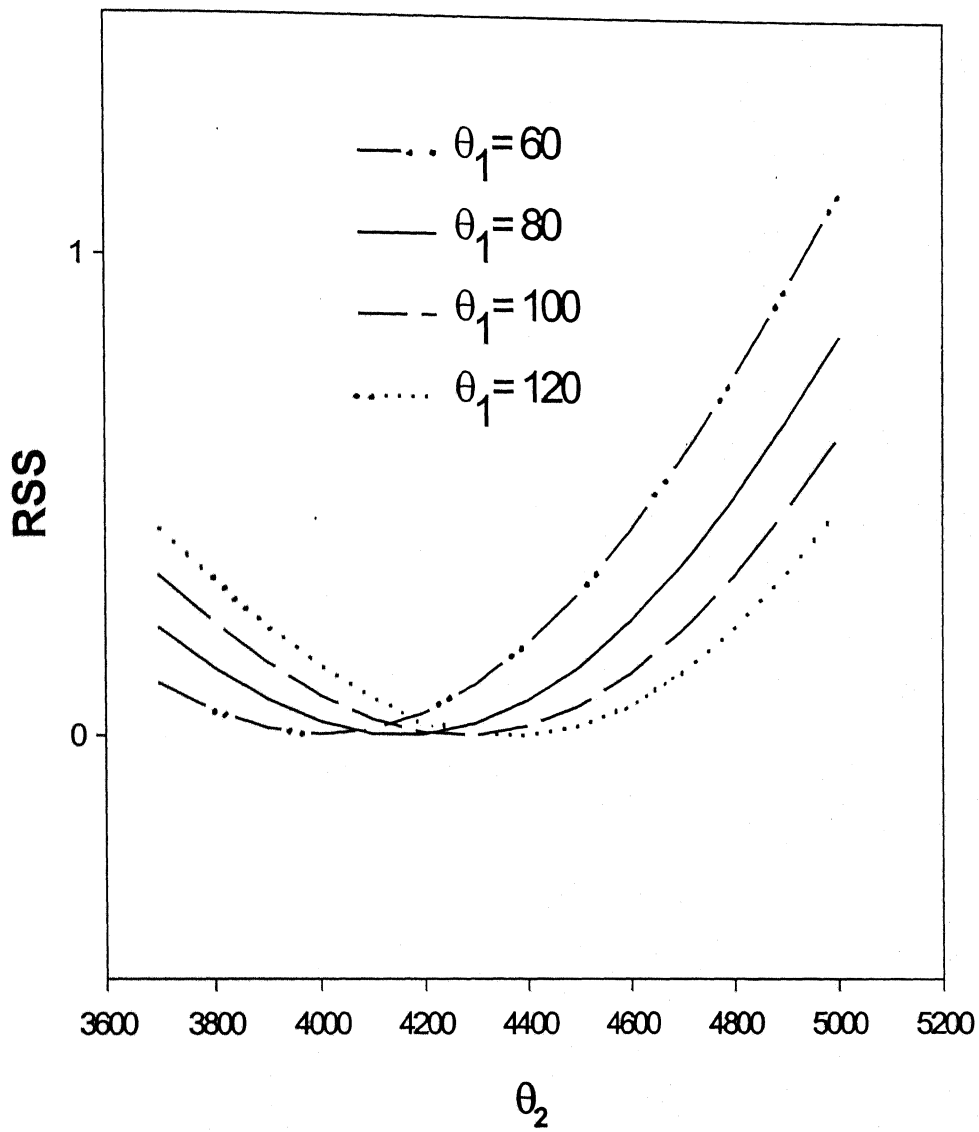


Fig. 4.3 Plots of RSS(residual sum of squares) vs θ_2

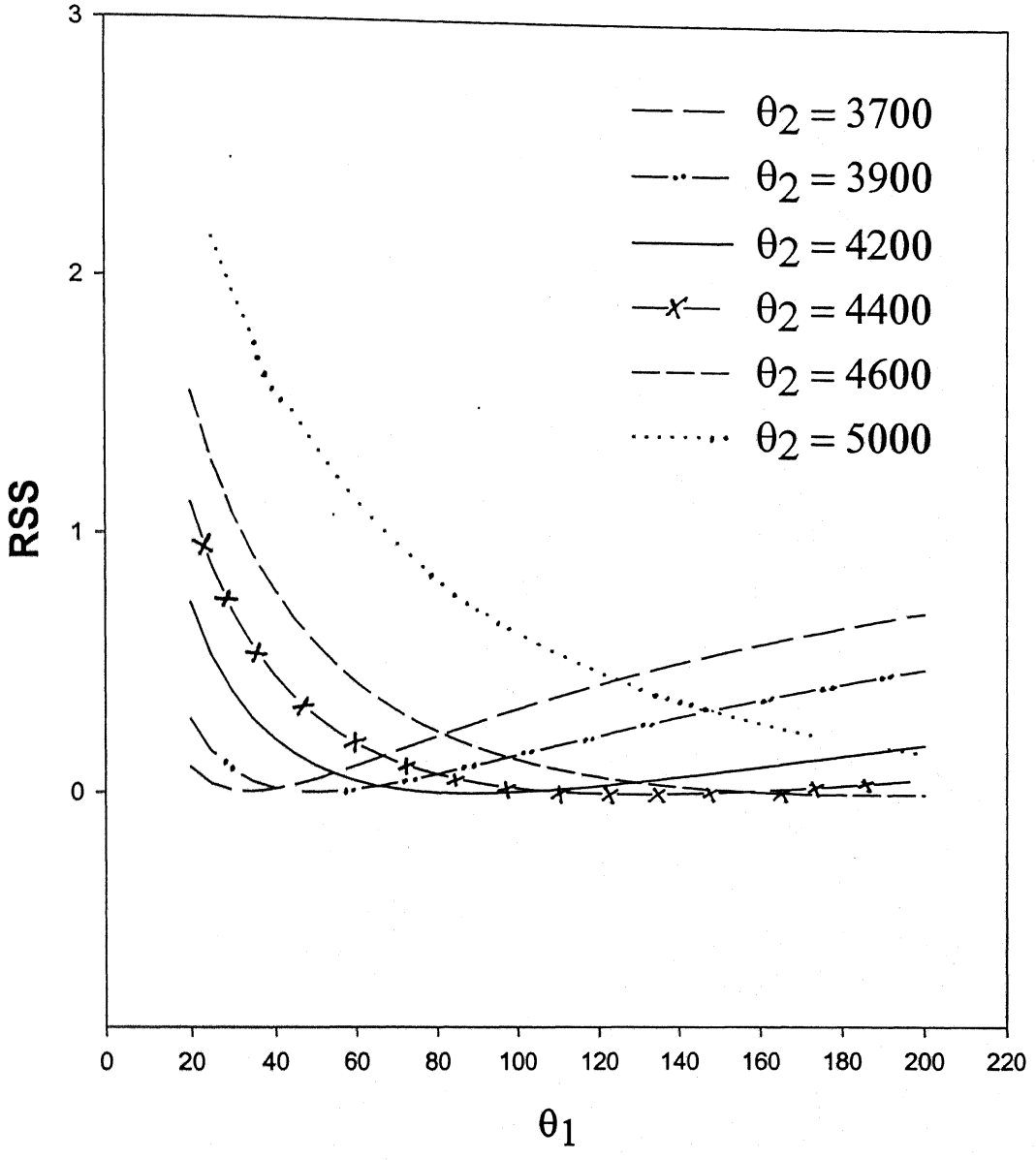


Fig 4.4 plots of RSS(residual sum of squares) vs θ_1

4.3.2 ESTIMATION OF MODEL PARAMETERS

The estimation of the model parameters has been done for model 2. From the Figures 4.3 and 4.4 it was observed that the minimum of the function lies in the range

$80 \leq \theta_1 \leq 120$

$4000 \leq \theta_2 \leq 4400$.

The starting estimates of the parameters for the *Marquardt* method are taken to be $\theta_1 = 100$ and $\theta_2 = 4000$ and for the *Simplex* method the starting simplex is [100,4000; 101,4000; 100,4001]. The convergence criteria chosen, for the *Simplex* method, is given by equation (3.7) and that for the *Marquardt* method it is given by equation (3.13). The estimation procedure is carried out using the two methods described in the previous chapter. The results obtained are presented in Table 4.2.

Table 4.2: Estimates of the parameters for model 2 by the *Simplex* and the *Marquardt* methods.

Method	Parameter θ_1	Parameter θ_2	No. of Iterations
Simplex Method	107.5640	4307.4150	94
Marquardt Method	105.9781	4300.448	1082

Taking the estimates of the parameters, for model 2, obtained by the *Marquardt* method the response data are calculated for model 2, which are given in Table 4.3 ($\sigma = 0$).

4.3.3 DATA GENERATION

To assess the reliability of the procedures used the data are generated by introducing random error into the simulated response, for model 2, and the model parameters are again calculated .The data are generated by the following procedure

$y_{new} = y_{old} + R \times \sigma$,

Where y_{new} are the data generated, y_{old} ($\sigma = 0$) are the values of the simulated response, R is some random number, which is in the range of –2 to +2, and σ is the standard deviation. The data generated for various σ values are given in Table 4.3.

Table 4.3: New Response data for Model 2.

Run	Input Variables		Response	Response	Response	Response
N	ξ_1	ξ_2	y_n $\sigma = 0.00$	y_n $\sigma = 0.001$	y_n $\sigma = 0.0001$	y_n $\sigma = 0.00001$
1	25	575	0.400575	0.398603	0.400381	0.400555
2	25	475	0.763409	0.763899	0.763470	0.763404
3	125	475	0.392224	0.392652	0.392451	0.392213
4	125	575	0.117896	0.115952	0.117833	0.117902
5	150	550	0.135301	0.134798	0.135126	0.135316
6	25	525	0.576695	0.576188	0.576901	0.576713
7	150	525	0.185044	0.184080	0.184982	0.185049
8	150	550	0.135301	0.134570	0.135287	0.135272
9	25	525	0.576695	0.576881	0.576816	0.576694

The estimates of the parameters are found by both the methods one thousand times. The means and the variances of the estimated parameters for different values of σ are given in the Tables 4.4 and 4.5 for the *Marquardt* and the *Simplex* methods respectively.

Table 4.4: The Means and the Variances of estimated parameters for Model 2 for various σ's by the *Marquardt* method.

σ	Mean (θ_1)	Mean (θ_2)	Variance (θ_1)	Variance (θ_2)
0.00	105.9781	4300.448	0.00	0.00
0.001	107.799	4308.674	3.933	92.74
0.0001	105.9850	4300.453	$1.015X10^{-4}$	$3.01X10^{-3}$
0.00001	105.9779	4300.485	$8.647X10^{-7}$	$1.50X10^{-3}$

Table 4.5: The Means and the Variances of estimated parameters for Model 2 for various σ's by the *Simplex* method.

σ	Mean (θ_1)	Mean (θ_2)	Variance (θ_1)	Variance (θ_2)
0.00	105.978	4300.446	0.00	0.00
0.001	108.5467	4312.238	5.01	133.50
0.0001	106.2296	4301.625	$5.26X10^{-2}$	1.31
0.0001	106.003	4300.561	$5.3X10^{-4}$	$1.32X10^{-2}$

The results in the above two tables show that the parameter estimates are approximately same, as the parameters at which we generated the response data, for $\sigma = 0$.

Now the parameters are estimated for all the other models by both the methods. The parameters estimated by *Simplex* method for different σ values are given in Table 4.6

Table 4.6: Estimates of the Model Parameters for other Models by the *Simplex* Method. (Model 2: True)

σ	Model (u)	($\theta_1^{(u)}$)	($\theta_2^{(u)}$)	No. of Iterations
0.00	1	25.0726	3813.622	125
	3	2280.649	5608.39	90
	4	129581.52	7390.389	227
0.001	1	25.407	3821.98	131
	3	not converged	not converged	--
	4	not converged	not converged	--
0.0001	1	24.469	3811.565	128
	3	2283.01	5610.781	89
	4	129799.756	7391.855	222
0.0001	1	25.034	3812.977	128
	3	2280.537	5610.781	69
	4	129581.52	7390.389	225

The parameters estimated for different σ values by the *Marquardt* method for all the other models are given in Table 4.7

Table 4.7: Estimates of the Model Parameters by the *Marquardt* method for other models. (Model 2: True)

σ	Model (u)	$(\theta_1^{(u)})$	$(\theta_2^{(u)})$	No. of Iterations
0.00	1	34.6	3978.693	1673
	3	2167.77	5582.11	1664
	4	126513.791	7346.683	1615
0.001	1	36.24	4000.504	1666
	3	2194.47	5588.674	1603
	4	129061.02	7419.22	1643
0.0001	1	34.7632	3980.915	1667
	3	2178.664	5584.558	1679
	4	129061.02	7376.72	1598
0.0001	1	34.598	3978.655	1681
	3	2165.137	5581.50	1598
	4	126910.19	7351.29	1587

The results in the Tables 4.6 and 4.7 show that the values of the parameters estimated from both the methods are quite close although the number of iterations done in the *Simplex* method is less as compared to the *Marquardt* method.

4.3.4 MODEL DISCRIMINATION

To see whether the model, from which we generated the data turns out to be true, the task of model discrimination has been performed. The *Bayesian* and the *Distance* methods are used and compared for model discrimination. The estimates of parameters obtained by both the methods of parameter estimation have been used for model discrimination. The prior probabilities associated with different models are chosen to be equal to 0.25. For the *Simplex* method, $\sigma = 0.001$, the parameters did not converge for models 3 and 4 so the discrimination is done only at the other two σ values.

The results of the *Distance* method applied for model discrimination using parameter values obtained by the *Simplex* method are shown in Table 4.8.

Table 4.8: Discrimination Among Rival Models: *Distance* Method. (Model 2 True)

σ	Discrimination Indices For Various Models			
	D^1	D^2	D^3	D^4
0.0001	0.3984	0.03917	0.208	0.3548
0.00001	0.3995	0.03871	0.201	0.3606

In the distance method the model, which has the smallest value of discrimination index is considered as the best one. The value of the discrimination index was calculated using the Equations 3.16 and 3.19.

From the above results it is explicit that the discrimination index for model 2, for different values of σ, is the least among all the rival models. This shows that model 2 is the best among the rival models.

The results of the *Distance* method applied for model discrimination using parameter values obtained by the *Marquardt* method are shown in Table 4.9.

Table 4.9: Discrimination Among Rival Models: *Distance* Method. (Model 2 True)

σ	Discrimination Indices For Various Models			
	D^1	D^2	D^3	D^4
0.001	.402	.0389	0.2142	0.3633
0.0001	.3949	0.0379	0.22	0.3564
0.00001	.389	0.0375	0.208	0.3647

From the above results it is explicit that model 2 has the lowest value of discrimination index for different values of σ. This shows that model 2 is the best among the rival models.

The results of the *Bayesian* method applied for model discrimination using parameter values obtained by the *Simplex* method are shown in Table 4.10

Table 4.10: Discrimination Among Rival Models: *Bayesian* Method. (Model 2 True)

σ	Posterior Probabilities For Various Models			
	P^1	P^2	P^3	P^4
0.0001	0.0000	0.9978	0.0014	0.0009
0.00001	0.0000	0.9981	0.0012	0.0008

Equation 3.14 was used to calculate the posterior probabilities of the rival models. The prior probability was chosen to be 0.25 for all the four models.

The results of the *Bayesian* method applied for model discrimination using parameter values obtained by the *Marquardt* method are shown in Table 4.11

Table 4.11: Discrimination Among Rival Models: *Bayesian* Method. (Model 2 True)

σ	Posterior Probabilities For Various Models			
	P^1	P^2	P^3	P^4
0.001	0.0001	0.9954	0.0027	0.0017
0.0001	0.0001	0.9966	0.0023	0.0010
0.00001	0.0000	0.9975	0.0018	0.0006

The results given in Tables 4.10 and 4.11 show that the posterior probability for model 2, for different values of σ, is the maximum. This shows that model 2 is the true model.

The work done for model 2 is now repeated assuming model 1 to be the true model. This is done to ensure that the method works properly for other models also.

The data are generated for model 1 with $\theta_1 = 50$ and $\theta_2 = 4000$ and $\sigma = 0.05$.

The generated data are given in Table 4.12.

TABLE 4.12: Response data using Model 1 as the correct model .

Run	Input Variables		Response
N	ξ_1	ξ_2	y_n
1	25	575	0.284707
2	25	475	0.759745
3	125	475	0.251773
4	125	575	0.005608
5	150	550	0.007334
6	25	525	0.541562
7	150	525	0.029464
8	150	550	0.002490
9	25	525	0.538896

The model parameters have been estimated for model 1 using both the methods (*Marquardt* and *Simplex*). The starting estimates of the parameters for the *Marquardt* method are taken to be $\theta_1 = 70$ and $\theta_2 = 4200$ and for the *Simplex* method the starting simplex is [70,4200; 71,4200; 70,4201]. The results are presented in Table 4.13.

Table 4.13: Estimates of the parameters for Model 1 by the *Simplex* and the *Marquardt* methods.

Method	Parameter θ_1	Parameter θ_2	No. of Iterations
Simplex Method	64.86	4127.765	96
Marquardt Method	65.4132	4132.261	824

Taking the estimates of the parameters, for model 1, obtained by *Marquardt* method the response data are again calculated for model 1, which are given in Table 4.14 ($\sigma = 0$). The data are now generated for various σ's which are as described below.

Table 4.14: New Response data for Model 1.

Run	Input Variables		Response	Response	Response	Response
N	ξ_1	ξ_2	y_n $\sigma = 0.00$	y_n $\sigma = 0.001$	y_n $\sigma = 0.0001$	y_n $\sigma = 0.00001$
1	25	575	0.290121	0.288849	0.289994	0.290108
2	25	475	0.761427	0.760265	0.761544	0.761438
3	125	475	0.255943	0.256261	0.255976	0.255927
4	125	575	0.002055	0.003398	0.002150	0.002060
5	150	550	0.004721	0.004738	0.004703	0.004735
6	25	525	0.535719	0.536210	0.535650	0.535731
7	150	525	0.023638	0.023962	0.023840	0.023640
8	150	550	0.004721	0.006127	0.004738	0.004711
9	25	525	0.535719	0.535800	0.535647	0.535729

The parameters are now estimated for all the other models by both the methods. The parameters estimated by *Simplex* method for different σ values are given in Table 4.15

Table 4.15: Estimates of the Model Parameters by the *Simplex* Method. (Model 1: True)

σ	Model (u)	$(\theta_1^{(u)})$	$(\theta_2^{(u)})$	No. of Iterations
0.00	1	65.41	4132.288	89
	2	790.427	5102.821	135
	3	84659.6	7202.89	201
	4	38402372.96	9929.60	352
0.001	1	66.9	4145.72	100
	2	807.247	5141.90	138
	3	not converged	not converged	--
	4	38966821.0	9945.30	340
0.0001	1	65.5	4133.60	100
	2	794.80	5132.70	140
	3	85219.0	7195.20	207
	4	38718502.1	9938.20	353
0.00001	1	65.4	4132.40	111
	2	791.7	5130.30	119
	3	84769.20	7195.7	213
	4	38496800.90	9936.32	338

The parameters estimated for different σ values by *Marquardt* method are given in Table 4.16.

Table 4.16: Estimates of the Model Parameters by the *Marquardt* Method. (Model 1: True)

σ	Model (u)	$(\theta_1^{(u)})$	$(\theta_2^{(u)})$	No. of Iterations
0.00	1	65.935	4132.375	829
	2	687.5234	5035.923	1970
	3	74619.61	7094.21	2012
	4	35061929.20	9791.4	1990
0.001	1	66.40	4140.64	799
	2	708.20	5073.11	1996
	3	79921.41	7205.42	1986
	4	37091456.73	9843.71	1989
0.0001	1	66.18	4138.16	873
	2	697.82	5065.20	1975
	3	77434.65	7144.18	1984
	4	36121567.16	9811.25	1994
0.0001	1	65.95	4136.53	804
	2	695.90	5063.78	1985
	3	75915.17	7102.13	2010
	4	35090143.54	9799.20	1996

The results in the above two tables show that the parameter estimates for model 1, obtained by both the methods, are approximately same as the parameters at which we generated the response data, for $\sigma = 0$. The parameter estimates for all the models at different σ values, obtained by both the methods, are also quite close. This shows that the both the methods work well and converge to nearly similar estimates although the number of iterations done in the *Simplex* method are less than as compared to the *Marquardt* method.

The model discrimination has been carried out using the estimates of parameters obtained by both the methods. The prior probabilities associated with different models are chosen to be equal to 0.25. The results of the *Distance* method applied for model discrimination for the *Simplex* method are shown in Table 4.17.

Table 4.17: Discrimination Among Rival Models: *Distance* Method. (Model 1 True)

σ	Discrimination Indices For Various Models			
	D^1	D^2	D^3	D^4
0.0001	0.0016	0.216	0.356	0.425
0.00001	0.0014	0.217	0.354	0.426

The above results show that the discrimination index for model 1, for different σ values, is the least. So model 1 is the best among the rival models.

The results of the distance method applied for model discrimination for the *Marquardt* method are shown in Table 4.18

Table 4.18: Discrimination Among Rival Models: *Distance* Method. (Model 1 True)

σ	Discrimination Indices For Various Models			
	D^1	D^2	D^3	D^4
0.001	0.0013	0.223	0.345	0.43
0.0001	0.0012	0.216	0.354	0.43
0.00001	0.0012	0.212	0.362	0.42

The above results show that the discrimination index for model 1, for different σ values, is the least. So model 1 is the best among the rival models.

The results of the *Bayesian* method applied for model discrimination for the *Simplex* method are shown in Table 4.19

Table 4.19: Discrimination Among Rival Models: *Bayesian* Method. (Model 1 True)

σ	Posterior Probabilities For Various Models			
	P^1	P^2	P^3	P^4
0.0001	0.9980	0.0019	0.0004	0.0000
0.00001	0.9981	0.0016	0.0002	0.0000

The results of the Bayesian method applied for model discrimination for the Marquardt method are shown in Table 4.20

Table 4.20: Discrimination Among Rival Models: *Bayesian* Method. (Model 1 True)

σ	Posterior Probabilities For Various Models			
	P^1	P^2	P^3	P^4
0.001	0.9973	0.0017	0.0009	0.0000
0.0001	0.9976	0.0015	0.0010	0.0000
0.00001	0.9978	0.0014	0.0008	0.0000

The results given in Tables 4.19 and 4.20 show that the posterior probability of model 1, for different σ values, is the maximum. This shows that model 1 is the best among the rival models.

4.4 DISCUSSION

4.4.1 COMPARISON OF THE METHODS USED FOR PARAMETER ESTIMATION

In the present work, the following two methods are being considered

- *Simplex method*
- *Marquardt method*

The *Simplex* method is based neither on gradients (first-order derivatives) nor on quadratic forms (second-order derivatives), where as the *Marquardt* method is basically an optimum interpolation between the Taylor series expansion and the gradient methods. *Marquardt* method shares with the gradient methods their ability to converge from an initial guess, which may be outside the region of convergence of other methods. On the basis of the results presented in the last few sections it is observed that the *Simplex* method is converging to the minimum value while doing less number of function evaluations. Both the methods don not converge when the value of σ is large, say 0.01, while for Simplex method, for model 3 and 4, it is not converging for $\sigma = 0.001$. The results shown in Table 4.4 and 4.5 show the mean and variance for 1000 iterations. It can be observed that the value of variance for both the methods at high σ values is more or less same but as we move towards the low σ values the variance, in the values of the parameters, differ quite a bit. The convergence obtained in the *Simplex* method is also faster as compared to the *Marquardt* method.

4.4.2 COMPARISON OF THE METHODS USED FOR MODEL DISCRIMINATION

The following two methods were used for model discrimination.

- *Bayesian method*
- *Distance method*

The parameters obtained by both the methods were used for model discrimination. The Posterior probability was obtained by the Bayesian method for different values of σ. The values of the posterior probability were found to be maximum for the model which was assumed to be true. The distance method was also used to discrimination among the rival models. In this method, the model, which has smaller discrimination index is considered as the best one. From the results of model discrimination it was observed that the discrimination index of the model which we assumed to be true is the least.

CHAPTER 5
SUMMARY AND CONCLUSIONS

In this dissertation the data are generated for the model and an experimental error is introduced into it. The parameters are estimated using two different methods and the methods are compared. The model discrimination based on the estimated parameters is also done by two different methods to show that the model assumed to be true initially arrives as the correct model finally.

One must choose between methods that need only evaluations of the function to be minimized and the methods that also require evaluation of the derivative of that function. The algorithms using the derivatives have more calculations.

A general problem encountered by all minimization methods is that of false convergence at a point other that the minimum. To avoid this problem the exhaustive search procedure was carried out to determine the range of the parameters in which the minimum value of the function lies. The estimates of the parameters obtained by both the methods viz, the Marquardt and the Simplex methods, for all the models and for different σ values converge in nearly the similar range although the number of function evaluations done in the *Simplex* method is less as compared to the *Marquardt* method. So the *Simplex* method is certainly better than the *Marquardt* method when one wants rapid convergence.

The methods used for model discrimination viz, the Bayesian method and the Distance method, were able to discriminate the rival models and arrive at the true model.

REFERENCES

Atkinson, A.C. and Cox, D.R. (1974). Planning experiments of discriminating between models. J.Roy.Soc.B, 36, 321-348.

Atkinson, A.C. and Donev, A.N. (1992). "Optimal experimental designs". Oxford University Press, New York, First Edition.

Bard, Y and Lapidus, L. (1968), Kinetics analysis by digital parameter estimation, Catalysis Rev., 2, 67-73.

Bates, D.M. and Watts, D.G. (1998). "Nonlinear regression analysis and its applications". New York Wiley.

Beck, J.V. and K.J. Arnold. (1977). "Parameter estimation in engineering and science". New York: Wiley.

Bryne, G.D. and Hall, C.A. (1973). "Numerical solutions of systems of nonlinear algebraic equations. Academic Press.

Box, G.E.P. and Hill, W.J. (1967). Discriminating among mechanistic models. Technometrics, 9, 57-71.

Box, G.E.P., Hunter, W.G. and Hunter, J.S. (1978). "Statistics for experimenters". New York: Wiley.

Chambers, J.M. (1973). Fitting nonlinear models: numerical techniques. Biometrika, 60, 1-13.

Draper, N.R. and Smith, H (1981). "Applied regression analysis". New York: Wiley, Second Edition.

Froment, G.F. (1975). Model discrimination and parameter estimation in heterogeneous catalysis. A.I.CH. E. J, 21, 1041-1057.

Hill, W.J. (1966). "Statistical techniques for model building". Ph.D. Thesis, University of Wisconsin.

Iyengar, S.S. and Rao, M.S. (1983). Statistical techniques in modeling of complex systems: single and multi response models. IEEE trans. On Sys: Man and cybernetics, SMC-13, 175-189.

Kuester, J.L. and Mize, J.H. (1973). "Optimization Techniques with Fortran".

Kittrell, J.R., Hunter, W.G., And Watson, C.C. (1966). Obtaining precise parameter estimates for nonlinear catalytic rate models. A.I.CH.E. J, 12, 1-5.

Lakshmi, V.M. (1999). "Comparison of various methods of model discrimination and design of experiments in single response systems". M.Tech. Thesis, I.I.T. Kanpur.

Lehmann, E.L. and Casella, G. (1998). "Theory of point estimation". Springer-Verlog, New York, Second Edition.

Marquardt, D.W. (1963). An algorithm for least squares estimation of nonlinear parameters, J. Soc. Ind. Appl. Math., 11, 431-441.

Mezaki, Kittrell and Watson. (1966). " Model building techniques for heterogeneous kinetics", British Chemical Engineering.

Nedler, J.A. and Mead, R. (1965). Computer Journal, 7, 308-313.

Prasad, K.B.S. (1976). "Use of expected likelihood in sequential model discrimination in multiresponse systems. Chem. Engg. Sc., 32, 1411-1418.

Press, W.H., Teukolsky, S.A., Vellerling, W.T. and Flannery, B.P. (1999). "Numerical recipes in C", second edition, Cambridge University Press.

Rohatgi, V.K. (1984). "Statistical Inference". New York Wiley.

Seinfeld, J.H. (1970). Nonlinear estimation theory, Ind. Engg. Chem., 62, 32-41.

Singh, S. and Rao, M. S. (1981). Parameter estimation and model discrimination in multi-response models. Indian Chem. Engineer, 13, 19-26.

Singh, S. (1986). "Design and analysis of experiments for model discrimination in uniresponse and multiresponse systems". Ph.D. Thesis, I.I.T. Kanpur.

APPENDIX 1

```
/* DETAILS ON THE SUBROUTINE AMOEBA CAN BE FOUND IN "NUMERICAL RECIPES IN C", PRESS ET. AL.(1999). */

                    /* PROGRAM FOR SIMPLEX METHOD */

# include <stdio.h>
# include <stdlib.h>
# include <math.h>
# include <conio.h>
# define NR_END 1
# define FREE_ARG char*
# define NMAX 5000
                /* Maximum allowed number of function evaluations. */
# define GET_PSUM \
                for ( j= 1 ; j<= ndim ; j++){\
                        for ( sum =0.0, i =1;i<=mpts ;i++)sum +=p[i][j];\
                                psum[j]=sum;}
# define SWAP(a,b) {swap=(a);(a)=(b);(b)=swap;}
void nrerror(char error_text[])
{
        printf("rajat");
        getchar();
        getchar();
        exit(1);
}

void free_vector(double *v, long nl, long nh)
{
        free((FREE_ARG) (v+nl-NR_END));
}

double *vector(long nl, long nh)
{
        double *v;
        v = (double *) malloc((size_t)((nh-nl+1+NR_END) * sizeof(double)));
        if (!v) nrerror("allocation failure in vector()");
        return v-nl+NR_END;
}

        /* START OF SUBROUTINE AMOEBA FOR SIMPLEX METHOD */
```

```
void amoeba(double **p,double y[],int ndim,double ftol,double(*funk)(double []),int
*nfunk)
{
        double amotry(double **p ,double y[],double psum[],int ndim,double
(*funk)(double []),int ihi, double fac);
        int i, ihi,ilo, inhi,j,mpts=ndim+1;
        double rtol,sum,swap,ysave,ytry,*psum;
        psum=vector(1,ndim);
        *nfunk=0;
        GET_PSUM
        for (;;){
                        ilo=1;
                        ihi=y[1]>y[2] ? (inhi=2,1): (inhi=1,2);
                        for (i=1;i<=mpts;i++)
                        {
                                if ( y [i]<=y[ilo]) ilo=i;
                                if( y[i]>y[ihi])
                                {
                                        inhi=ihi;
                                        ihi=i;
                                }else if(y[i]>y[inhi] && i!=ihi) inhi=i;
                        }
                        rtol =2.0*fabs(y[ihi]-y[ilo])/(fabs(y[ihi])+fabs(y[ilo]));
                        if (rtol<ftol)
                        {
                                SWAP(y[1],y[ilo])
                                for (i=1;i<=ndim;i++)
                                SWAP(p[1][i],p[ilo][i])
                                break;
                        }
                        if( *nfunk>=NMAX) nrerror("NMAX exceeded");
                        *nfunk +=2;
                        ytry = amotry(p,y,psum,ndim,funk,ihi,-1.0);
                        if (ytry<=y[ilo])
                        ytry = amotry(p,y,psum,ndim,funk,ihi,2.0);
                        else if (ytry>y[inhi])
                        {
                                ysave = y[ihi];
          ytry=amotry(p,y,psum,ndim,funk,ihi,0.5);
                                if (ytry>=ysave)
                                {
                                        for (i =1 ; i<=mpts;i++)
                                        {
                                                if (i!= ilo)
                                                {
                                                        for ( j=1;j<=ndim;j++)
                                                                p[i][j]
                                                                        =
psum[j]=0.5*(p[i][j]+p[ilo][j]);
                                                        y[i]=(*funk)(psum);
```

```
                                        }
                                    }
                                    *nfunk += ndim;
                                    GET_PSUM
                                }
                            } else --(*nfunk);
                        }
                        free_vector(psum,1,ndim);
                }
double amotry(double **p ,double y[],double psum[],int ndim,double (*funk)(double
[]),int ihi, double fac)
{
        int j;
        double fac1,fac2,ytry,*ptry;
        ptry= vector (1,ndim);
        fac1=(1.0-fac)/ndim;
        fac2=fac1-fac;
        for( j=1;j<=ndim; j++) ptry[j]=psum[j]*fac1-p[ihi][j]*fac2;
        ytry=(*funk)(ptry);
        if (ytry<y[ihi]){
                y[ihi]=ytry;
                for (j=1;j<=ndim;j++)
                {
                        psum[j] += ptry [j] -p[ihi][j];
                        p[ihi][j]=ptry[j];
                }
        }
        free_vector (ptry,1,ndim);
        return ytry;
}
                        /* END OF SUBROUITNE AMOEBA */

                /* THE OBJECTIVE FUNCTION IS DEFINED AS funk */

double funk(double x[])
{
                FILE *in1, *in2;
                double x1[10],x2[10],y[10],yhat,de2,sum,g,r[10],y1[10];
                int i;
                in1 = fopen("dat1.txt","r");

/* dat1.txt IS THE FILE THAT CONTAINS THE VALUES OF DFEPENDENT AND
   INDEPENDENT VARIABLES . SINCE THERE ARE 9 DATA POINTS SO i IS
                        VARIED UPTO 9 IN THE FOR LOOP. */

                for(i=1;i<=9;i++)
                {
                        fscanf(in1,"%lf %lf %lf",&x1[i],&x2[i],&y[i]);
        }
                in2 = fopen("random.dat","r");
```

```
/* random.dat IS THE FILE THAT CONTAINS THE RANDOM NUMBERS. THE
    RANDOM ERROR IS ADDED TO THE PREDICTED RESPONSE y[i].*/

            for(i=1;i<=9;i++)
            {
                    fscanf( in2,"%lf",&r[i]);

                    y1[i]=y[i]+0.0001*r[i];
            }
      fclose(in2);
            sum =0.0;
            for(i=1; i<=9;i++)
            {
   /* yhat IS THE RESPONSE OF THE MODEL TAKEN. THE OBJECTIVE
FUNCTION de2  IS CALCULATED BELOW AND THE VALUE IS STORED IN g
    AND THEN THIS VALUE IS RETURNED TO THE FUNCTION funk. */
 /* IN THE MODEL x[1] and x[2] ARE THE PARAMETERS WITH RESPECT O
 WHICH THE MINIMIZATION OF THE FUNCTION IS DONE AND x1[i], x2[i]
                ARE THE INDEPENDENT VARIABLES. */

      yhat =1.0/(1.0+x[1]*x1[i]*exp(-x[2]/x2[i]));
                    de2 =pow ((y1[i]-yhat),2);
                    sum += de2;
            }
            g = sum;

            fclose(in1);
            return (g);
}

            /* MAIN PROGRAM FOR THE SIMPLEX METHOD */

void main()
{

      FILE *in,*out,*out1;

     /* BELOW IS THE INITIAL GUESS PROVIDED FOR THE MODEL */

       double q[4][3]={{0.0,0.0,0.0},
                                {0.0,100,4000},
                                {0.0,101,4000},
                                {0.0,100,4001}
                                };

       double *p[3];

       double y[4];
       int *nfunk;
```

```
        int i,j,k;
        float dummy;
        nfunk = (int *) malloc(1*sizeof(int));
        in = fopen("ran1.dat","r");
        for(i=0; i<4; i++)
        {
                p[i]=(double *) malloc(3*sizeof(double));
                for(j=0; j<3; j++)
                {
                        *(*(p+i)+j)=q[i][j];
                }
        }
/* THE FOR LOOP APPLIED BELOW COMPUTES THE PARAMETERS FOR 1000
                                        TIMES */
        for(i=0; i<1000; i++)
        {
  /* Y[] ARE THE OUTPUT VALUES OF THE OBJECTIVE FUNTION AT THE
                                        INITIAL GUESSES */
                y[0]=0.0;y[1]= 0.084925;y[2]=0.087956;y[3]=0.084340;

               /* OUTPUT IS BIENG STORED IN THE FILE reslt.res */

                out = fopen("random.dat","w");
                out1 = fopen("reslt.res","a+");
                for(j=0; j<9; j++)
                {
                        fscanf(in," %e",&dummy);
                        printf("%f",dummy);
                        fprintf(out," \n %f",dummy);
                }
                fclose(out);

  /* CALL OF SUBROUTINE AMOEBA IS MADE  AND WITHIN AMOEBA THE
                                FUNCTION funk IS ALSO CALLED.*/

                amoeba(p,y,2,0.000001,funk,nfunk);
        for(k=1; k<4; k++)
        {
                for(j=1; j<3; j++)
                {
                        fprintf(out1,"\n\t%f", *(*(p+k)+j));
                }
        }
/* THE NUMBER OF FUNCTION AVLUATIONS IS CALCULATED BELOW AND
                                STORED IN THE FILE reslt.res */
        fprintf(out1,"\n%d",*nfunk);
        fclose(out1);
        }
}
                        /* END OF THE MAIN PROGRAM */
```

APPENDIX II

The following subroutine, Kuester and Mize (1973), was used for parameter estimation for the *Marquardt* method.

```
C     MAINLINE PROGRAM FOR SUBROUTINE BSOLVE

C     RANDOM NUMBER GENERATION SUBROUTINE RNSET AND
C     RNNOR IS TAKEN FROM IMSL LIBRARY

        USE MSIMSLMS
        USE MSIMSLSS

      DIMENSION P(50),A(10,10),AC(10,10),X1(10),X2(10)
      DIMENSION B(10),Z(10),Y(10),BV(10),BMIN(10),BMAX(10)
      DIMENSION Y0(10)
      INTEGER :: M
      INTEGER   ISEED, NOUT, NR
      EXTERNAL FUNC
      COMMON X1,X2
      REAL    R(9)
      ISEED = 0
C     THE DO LOOP BELOW CALCULATES THE ESTIMATES OF THE
C     PARAMETERS FOR THE REQUIRED NUMBER OF TIMES.
      DO M = 1,1
C     FILE INPUT.DAT CONTAINS THE VALUES OF
      NN,KK,BMIN,BMAX,X1,X2.
C     FILE YIN.DAT CONTAINS THE DATA FOR Y(RESPONSE).
C     FILE DOCUMENP.DAT IS THE OUTPUT FILE.
C     FILE YVALUES.DAT CONTAINS THE NEW RESPONSE VALUE AFTER
C     EACH ITERATION.
        NI1 =2
        NI = 50
        NO = 66
        NY = 3
        OPEN(UNIT = NI, FILE = 'INPUT.DAT')
        OPEN(UNIT = NI1,FILE = 'YIN.DAT')
        OPEN(UNIT = NO, FILE = 'DOCUMENTP.DAT')
        OPEN(UNIT = NY, FILE = 'YVALUES.DAT')
C
C       READ IN NUMBER OF DATA POINTS, UNKNOWNS.
```

```
C
      READ (NI,*) NN, KK

C
C     READ IN INITIAL GUESSES.
C
      READ (NI,*) (B(J),J=1,KK)

C
C     READ IN LIMITS OF VARIABLE.
C
      READ (NI,*) (BMIN(J),J=1,KK)
      READ (NI,*) (BMAX(J),J=1,KK)

C     READ IN INDEPENDENT VARIABLES.
C     READ IN DEPENDENT VARIABLES.
C
      READ (NI,*) (X1(I),I=1,NN)
      READ (NI,*) (X2(I),I=1,NN)
      READ (NI1,*) (Y0(I),I=1,NN)
      REWIND(NI1)
      CLOSE(NI1)
C     RANDOM NO. GENERATES HERE
      NR   = NN
      CALL RNSET (ISEED)
      CALL RNNOR (NR, R)
      OPEN (1,FILE="RANDOM.DAT")
      WRITE(1,*) R
      ISEED=ISEED + 20

      WRITE(NY,*)
      DO I = 1,NN
      Y(I) = Y0(I) + 0.00*R(I)
      WRITE(NY,*) Y(I)
      END DO

C
      FNU = 0.0
      FLA = 1.0
      TAU = 0.0
      EPS = 0.0
      PHMIN = 0.0
      I = 0
      KD = KK
      FV = 0.0
      DO 100 J=1,KK
      BV(J)=1
100   CONTINUE
      ICON = KK
```

```
          ITER = 0
          WRITE (NO,*)

!015      FORMAT (1H1,10X,27HBSOLVE REGRESSION ALGORITHM )
C         THE SUBROUTINE BSOLVE IS CALLED HERE

200       CALL  BSOLVE(KK,B,NN,Z,Y,PH,FNU,FLA,TAU,EPS,PHMIN
    1     ,I,ICON,FV,DV,BV,BMIN,BMAX,P,FUNC,DERIV,KD,A,AC,GAMM)
C
          ITER = ITER + 1
         WRITE (NO,*) ICON,PH,ITER
!001     FORMAT (/,2X,6HICON = ,I3,4X,5HPH = ,E15.8,4X,
  !1     16HITERATION NO. = ,I3)
          IF (ICON) 10, 300,200
10        IF (ICON + 1) 20,60,200
20        IF (ICON + 2) 30,70,200
30        IF (ICON + 3) 40,80,200
40        IF (ICON + 4) 50,90,200
50        GO TO 95
60        WRITE (NO,*)
!004      FORMAT (//,2X,32HNO FUNCTION IMPROVEMENT POSSIBLE)
          GO TO 300
70        WRITE (NO,*)
!005      FORMAT (//,2X,28HMORE UNKNOWNS THAN FUNCTIONS )
          GO TO 300
80        WRITE (NO,*)
!006      FORMAT (//,2X,24HTOTAL VARIABLES ARE ZERO)
          GO TO 300
90        WRITE (NO,*)
!007      FORMAT (//,2X,79HCORRECTIONS SATISFY CONVERGENCE
!   1     REQUIREMENTS BUT LAMDA FACTOR (FLA) STILL LARGE)
          GO TO 300
95        WRITE (NO,*)
!008      FORMAT (//,2X,20HTHIS IS NOT POSSIBLE)
          GO TO 300
300       WRITE (NO,*)
!002      FORMAT (//,2X,26HSOLUTIONS OF THE EQUATIONS)
          DO 400 J=1,KK
          WRITE (NO,*)  B(J)

          WRITE(*,*) J,B(J)

!003      FORMAT (/,2X, 2HB(,I2,4H) = ,E16.8)
400       CONTINUE
          CLOSE(NI)
          WRITE(*,*) 'M = ',M
          END DO
          CLOSE(1)
          CLOSE(NY)
          CLOSE(NO)
```

```
1000    STOP
        END

C       THE USER SUPPLIED FUNCTION IS GIVEN BELOW

        SUBROUTINE FUNC (KK, B, NN, Z, FV)
C
        DIMENSION X1(10),X2(10)         , Z(25), B(25)
        COMMON X1,X2

C       Z(JJ) REPRESENTS THE MODEL, B(1),B(2)ARE THE UNKNOWN
C       PARAMETERS AND X1(JJ) AND X2(JJ) ARE THE INDEPENDENT
C       VARIABLES
        DO 100 JJ=1,NN
        Z(JJ) = 1/(1+B(1)*X1(JJ)*EXP(-B(2)/X2(JJ)))

100     CONTINUE

C
        RETURN
        END

C       SUBROUTINE BSOLVE IS DEFINED BELOW

        SUBROUTINE  BSOLVE (KK,B,NN,Z,Y,PH,FNU,FLA,TAU,EPS,
     1  PHMIN,I,ICON,FV,DV,BV,BMIN,BMAX,P,
     2  FUNC,DERIV, KD,A,AC,GAMM)
        DIMENSION B(10),Z(10),Y(10),BV(10),BMIN(10),BMAX(10)
        DIMENSION P(50),A(10,10),AC(10,10),X(10),FV(10),DV(10)
C
        K=KK
        N=NN
        KP1= K+1
        KP2= KP1 + 1
        KBI1= K*N
        KBI2= KBI1 + K
        KZI=KBI2 + K
        IF (FNU .LE. 0.) FNU = 10.0
        IF (FLA .LE. 0.) FLA = 0.01
        IF (TAU .LE. 0.) TAU = 0.001
        IF (EPS .LE. 0.) EPS = 0.00002
        IF (PHMIN.LE.0.) PHMIN =0.
120     KE=0
130     DO 160 I1 =1,K
160     IF (BV(I1) .NE. 0. ) KE = KE + 1
        IF (KE .GT. 0)  GO TO 170
162     ICON = -3
163     GO TO 2120
170     IF (N .GE. KE ) GO TO 500
```

```
180   ICON = -2
190   GO TO 2120
500    I1 =1
530    IF (I .GT. 0 ) GO TO 1530
550    DO 560 J1=1,K
       J2 = KBI1 + J1
       P(J2) = B(J1)
       J3 = KBI2 + J1
560    P(J3) = ABS(B(J1)) + 1.0E-02
       GO TO 1030
590    IF (PHMIN .GT. PH .AND. I .GT. 1) GO TO 625
       DO 620 J1=1,K
       N1 = (J1-1)*N
       IF (BV(J1) ) 601,620,605
601    CALL DERIV (K,B,N,Z,P(N1+1),FV,DV,J1,JTEST)
       IF (JTEST .NE. (-1)) GO TO 620
       BV (J1) = 1.0
605    DO 606 J2 =1,K
       J3 = KBI1 + J2
606    P(J3) = B(J2)
       J3 = KBI1 + J1
       J4 = KBI2 + J1
       DEN = 0.001*AMAX1(P(J4),ABS(P(J3)))
       IF (P(J3) + DEN .LE. BMAX(J1)) GO TO 55
       P(J3) = P(J3) - DEN
       DEN   = -DEN
       GO TO 56
55     P(J3) = P(J3) + DEN
56     CALL FUNC(K,P(KBI1+1),N,P(N1+1),FV)

        DO 610 J2=1,N
        JB  = J2 +N1
610     P(JB) = (P(JB) - Z(J2))/DEN
620     CONTINUE
C
C       SET UP CORRECTION EQUATIONS
C
625     DO 725 J1=1,K
        N1 = (J1-1)*N
        A(J1,KP1) = 0.
        IF (BV(J1) ) 630,692,630
630     DO 640 J2=1,N
        N2 = N1 +J2
640     A(J1,KP1) = A(J1,KP1) + P(N2)*(Y(J2)-Z(J2))

650     DO 680 J2=1,K
660     A(J1,J2)=0.
665     N2 = (J2-1)*N
670     DO 680 J3=1,N
672     N3 = N1 + J3
```

```
674     N4 = N2 + J3
680     A(J1,J2)=A(J1,J2)+P(N3)*P(N4)

        IF (A(J1,J1).GT.1.E-20) GO TO 725
692     DO 694 J2 =1 ,KP1
694     A(J1,J2)=0.0

695             A(J1,J1) = 1.0

725             CONTINUE
                GN =0.
                DO 729 J1=1,K

729             GN = GN +A(J1,K)**2

C
C               SCALE CORRECTION EQUATIONS
C
                DO 726 J1=1,K
726             A(J1,KP2) =  SQRT(A(J1,J1))

                DO 727 J1=1,K
                A(J1,KP1)=A(J1,KP1)/A(J1,KP2)
                DO 727 J2 =1,K
727             A(J1,J2) = A(J1,J2)/(A(J1,KP2)*A(J2,KP2))

730             FL = FLA/FNU

                GO TO 810
800             FL = FNU*FL

810             DO 840 J1=1,K
820             DO 830 J2=1,KP1
830             AC(J1,J2)=A(J1,J2)
                AC(J1,J1)=2

840             CONTINUE
C
C               SOLVE CORRECTION EQUATIONS
C
                DO 930 L1=1,K
                L2 = L1 +1
                DO 910 L3=L2,KP1
910             AC(L1,L3)=AC(L1,L3)/AC(L1,L1)
                DO 930 L3=1,K
                IF (L1-L3) 920,930,920
920             DO 925 L4=L2,KP1
925             AC(L3,L4)=AC(L3,L4)-AC(L1,L4)*AC(L3,L1)

930             CONTINUE
```

```
C
                DN = 0.0
                DG = 0.0
                DO 1028 J1=1,K
                AC (J1,KP2) = AC(J1,KP1)/A(J1,KP2)

                J2 = KBI1 + J1
              P(J2) = AMAX1(BMIN(J1),AMIN1(BMAX(J1),B(J1)+AC(J1,KP2)))
              DG = DG + AC(J1,KP2)*A(J1,KP1)*A(J1,KP2)
              DN = DN + AC(J1,KP2)*AC(J1,KP2)
1028          AC (J1,KP2) = P(J2)-B(J1)

              COSG  = DG/SQRT (DN*GN)

              JGAM = 0
              IF (COSG) 1100,1110,1110
1100          JGAM=2
             COSG = -COSG
1110          CONTINUE
              COSG  = AMIN1(COSG,1.0)
              GAMM = ARCOS(COSG)*180./(3.14159265)
              IF (JGAM .GT. 0 ) GAMM = 180. -GAMM
1030          CALL FUNC   (K,P(KBI1+1),N,P(KZI+1),FV)
1500          PHI = 0.
              DO 1520 J1=1,N
              J2 = KZI + J1
1520          PHI=PHI+(P(J2)-Y(J1))**2
              IF (PHI.LT. 1.E-10) GO TO 3000
              IF (I .GT. 0 ) GO TO 1540
1521          ICON  = K
              GO TO 2110
1540          IF (PHI .GE. PH ) GO TO 1530
C
C             EPSILON TEST
C
1200          ICON = 0
              DO 1220 J1=1,K
              J2= KBI1+J1
1220          IF (ABS(AC(J1,KP2))/(TAU+ABS(P(J2))) .GT. EPS ) ICON = ICON +1
              IF (ICON .EQ. 0) GO TO 1400
C
C             GAMMA LAMBDA TEST
C
              IF (FL .GT. 1.0 .AND. GAMM .GT. 90.0) ICON =-1
              GO TO 2105
C
C             GAMMA EPSILON TEST
C
1400          IF (FL .GT. 1.0 .AND. GAMM .LE. 45.0) ICON =-4
             GO TO 2105
```

```
C
1530    IF (I1-2) 1531,1531,2310
1531    I1=I1+1
        GO TO (530,590,800),I1
2310    IF(FL .LT. 1.0E+8) GO TO 800
1320    ICON =-1
C
2105            FLA =FL
                DO 2091 J2 =1,K
                J3 = KBI1 +J2
2091            B(J2)=P(J3)
2110            DO 2050 J2=1,N
                J3 = KZI+J2
2050            Z(J2) = P(J3)
                PH =PHI
                I= I+1
2120            RETURN
3000            ICON = 0
                GO TO 2105
C
                END


                FUNCTION ARCOS(Z)
C
                X=Z
                KEY=0
                IF( X.LT. (-1.)) X=-1.
                IF( X.GT. 1.)   X=1.
                IF( X.GE. (-1.)         .AND. X .LT. 0.) KEY=1
                IF(X.LT. 0.) X=ABS(X)
                IF ( X.EQ. 0. ) GO TO 10
                ARCOS = ATAN (SQRT(1.-X*X)/X)
                IF (KEY .EQ. 1) ARCOS=3.14159265-ARCOS
                GO TO 999
10              ARCOS = 1.5707963
C
999             RETURN
                END
```